Practical Electronics

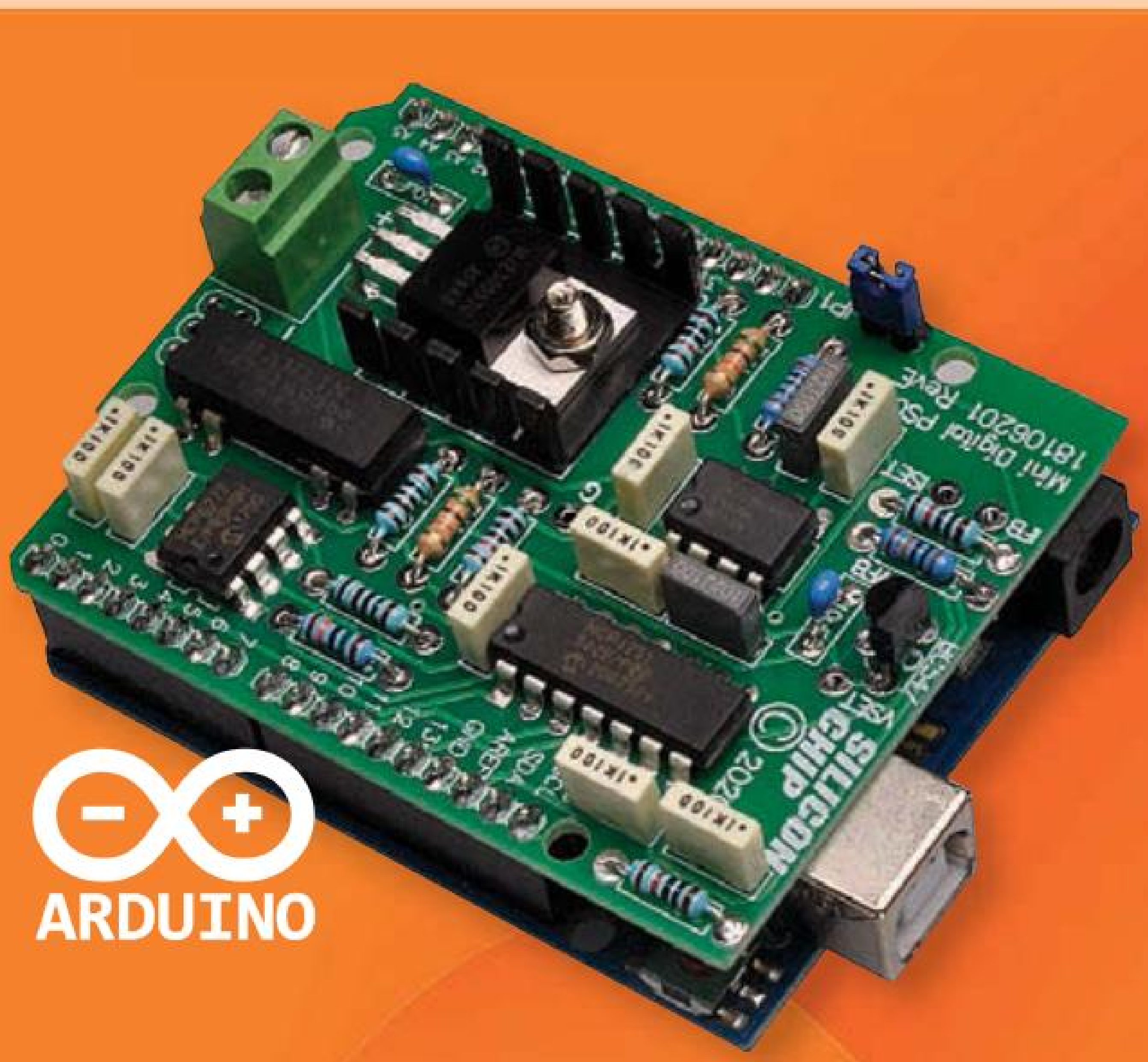
The UK's premier electronics and computing maker magazine

Circuit Surgery
Exploring op amp input offsets

Audio Out
Vocoder: Driver Amplifier build

KickStart
Using the I^{2}C bus

Make it with Micromite
Using and interfacing the versatile iButton


ARDUINO
SILICON CHIP
Mini Digital PSU
18106201 Rev B

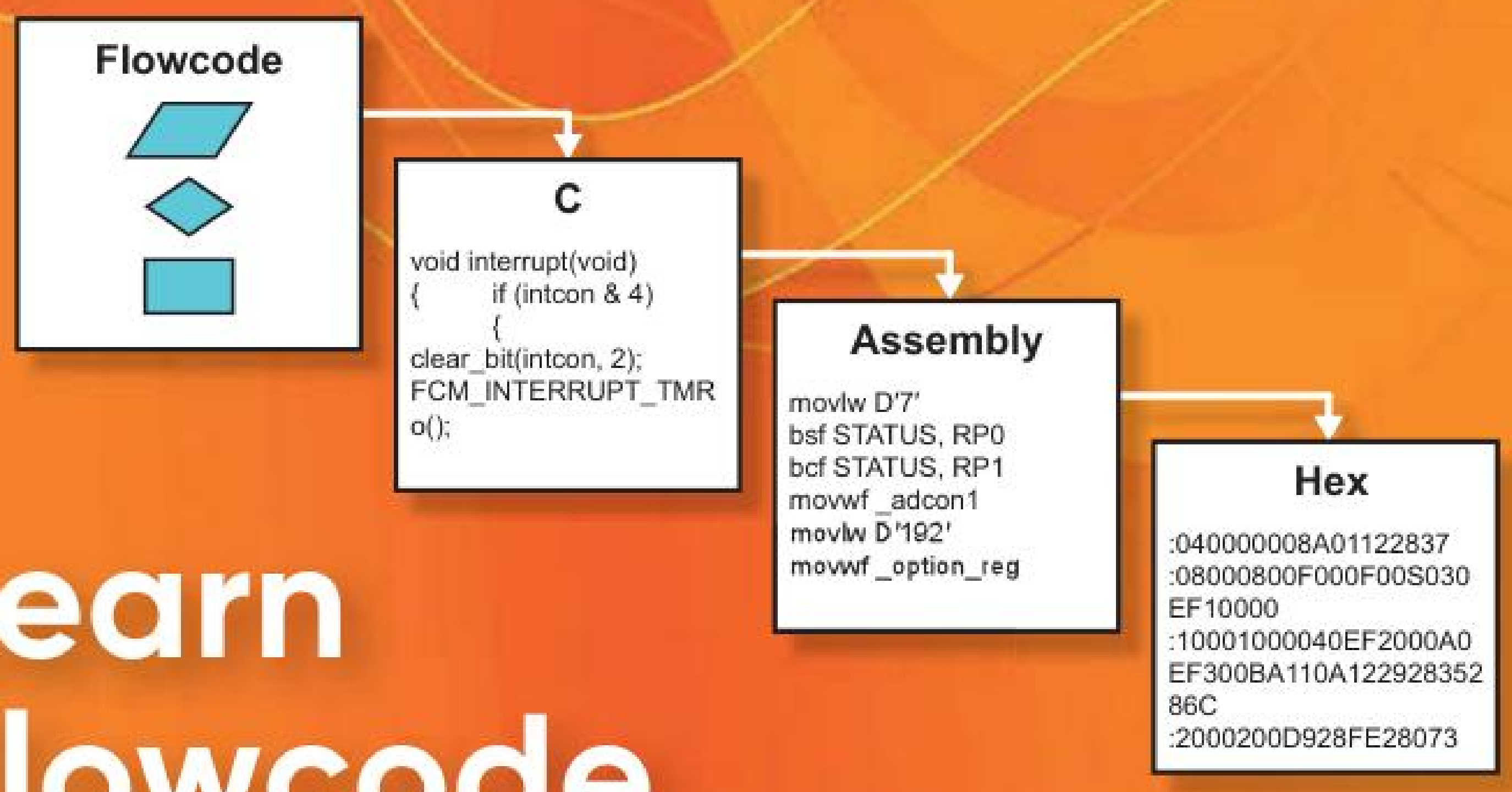
237.2 V
6.32 A
1502.0 W
50.0 Hz
0.10 kWh
1.00 PF

Mastering AC meters

Build this handy Arduino-based power supply

Soothing Electronic Wind Chimes

WIN!
Microchip WLR089 Xplained Pro Evaluation Kit
WIN!


Flowcode
C
Assembly
Hex

```
void interrupt(void)
{         if (intcon & 4)
          {
clear_bit(intcon, 2);
FCM_INTERRUPT_TMR
o();
```

```
movlw D'7'
bsf STATUS, RP0
bcf STATUS, RP1
movwf _adcon1
movlw D'192'
movwf _option_reg
```

```
:040000008A01122837
:08000800F000F00S030
EF10000
:10001000040EF2000A0
EF300BA110A122928352
86C
:2000200D928FE28073
```

Learn Flowcode Programming

DATA
Export
14.4
5
I1
I2
I3
I4
-5
-48 hours
Hours
Days
SILICON CHIP © 2020
Smart Battery Monitor and Logger

Battery Monitor Logger

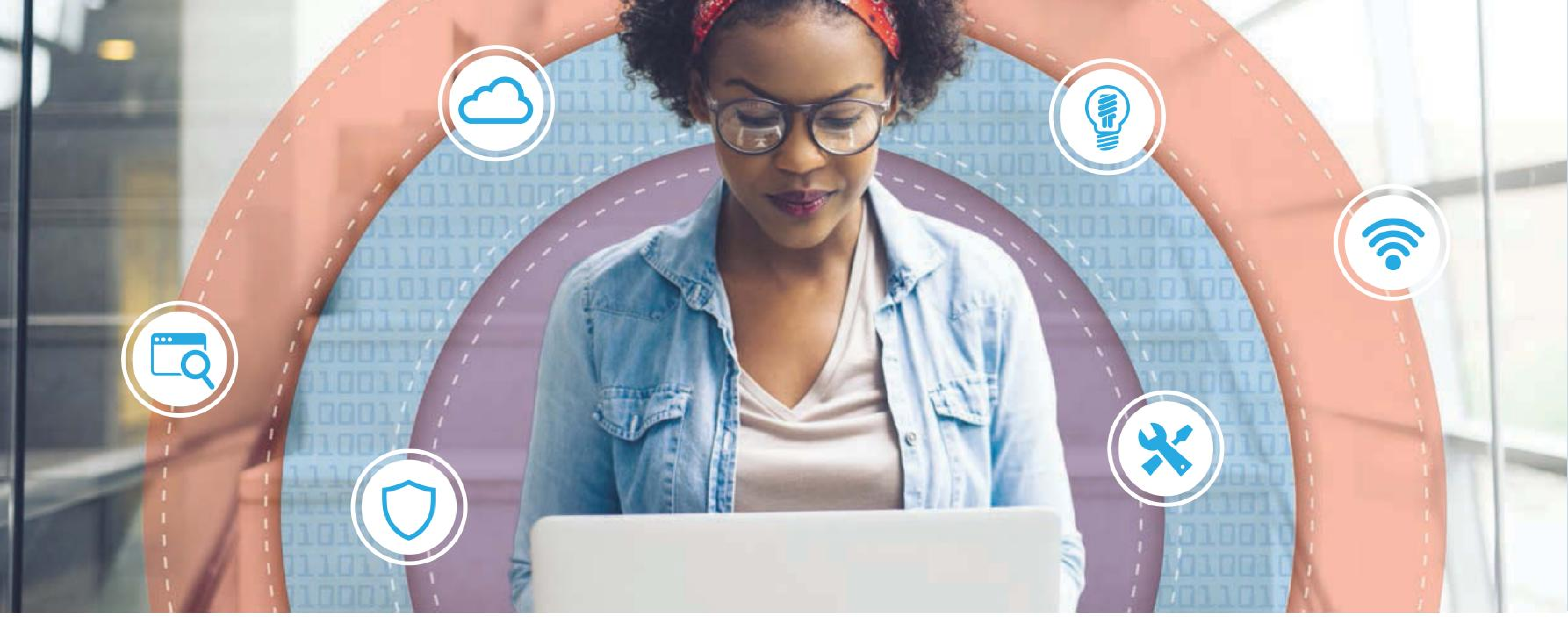
PLUS!

Techno Talk – Should we be worried?

Net Work – Electricity generation and streaming radio

Feb 2022 £5.49

www.electronpublishing.com | @practicalelec | practicalelectronics



MPLAB® Cloud Tools Ecosystem

Discover, Configure and Develop: An Ecosystem for All Your Ideas

MPLAB cloud tools ecosystem is a complete online solution for all skill-level users, to discover, configure, develop and debug embedded PIC® and AVR® microcontroller (MCU) applications.

- Intuitive entry into PIC and AVR MCU development with integrated MPLAB tools
- Quick prototyping with Microchip's PIC and AVR MCUs using Curiosity boards
- No software installation to start developing PIC and AVR MCU solutions

Designers who use PIC and AVR MCUs can visit the MPLAB Cloud Tools page and start developing based on their design phase.

- Search and Discovery: Access MPLAB Discover to find fully configured and complete source code projects
- Configure Code: Easily configure software applications with MPLAB Code Configuration
- Develop and Debug: Developing, debugging and deploying project applications directly from a preferred browser can be completed without any software installation with MPLAB Xpress IDE

MPLAB® DISCOVER

MPLAB® CODE CONFIGURATOR

MPLAB® XPRESS IDE

MICROCHIP

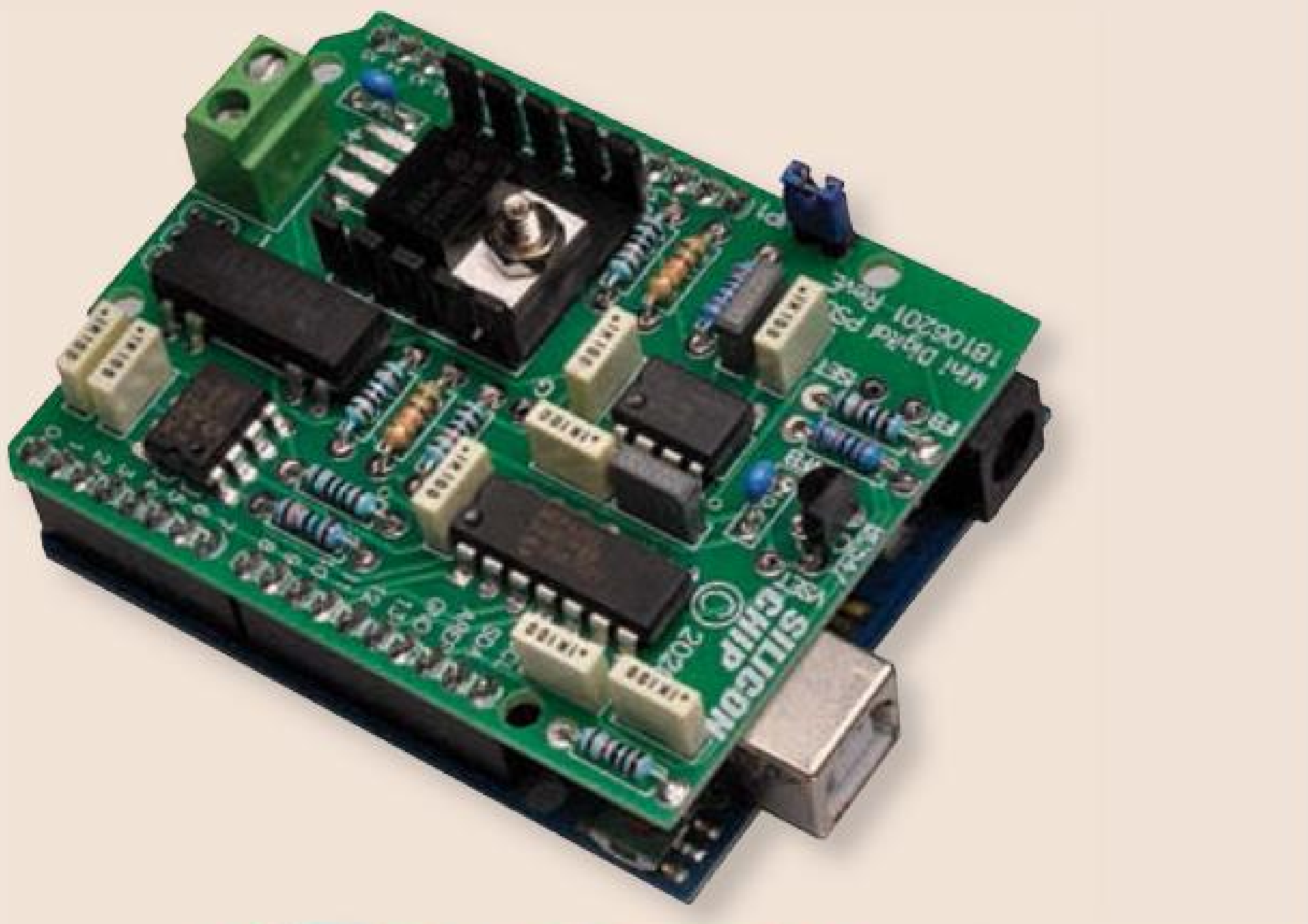
microchip.com/MPLABcloudtools

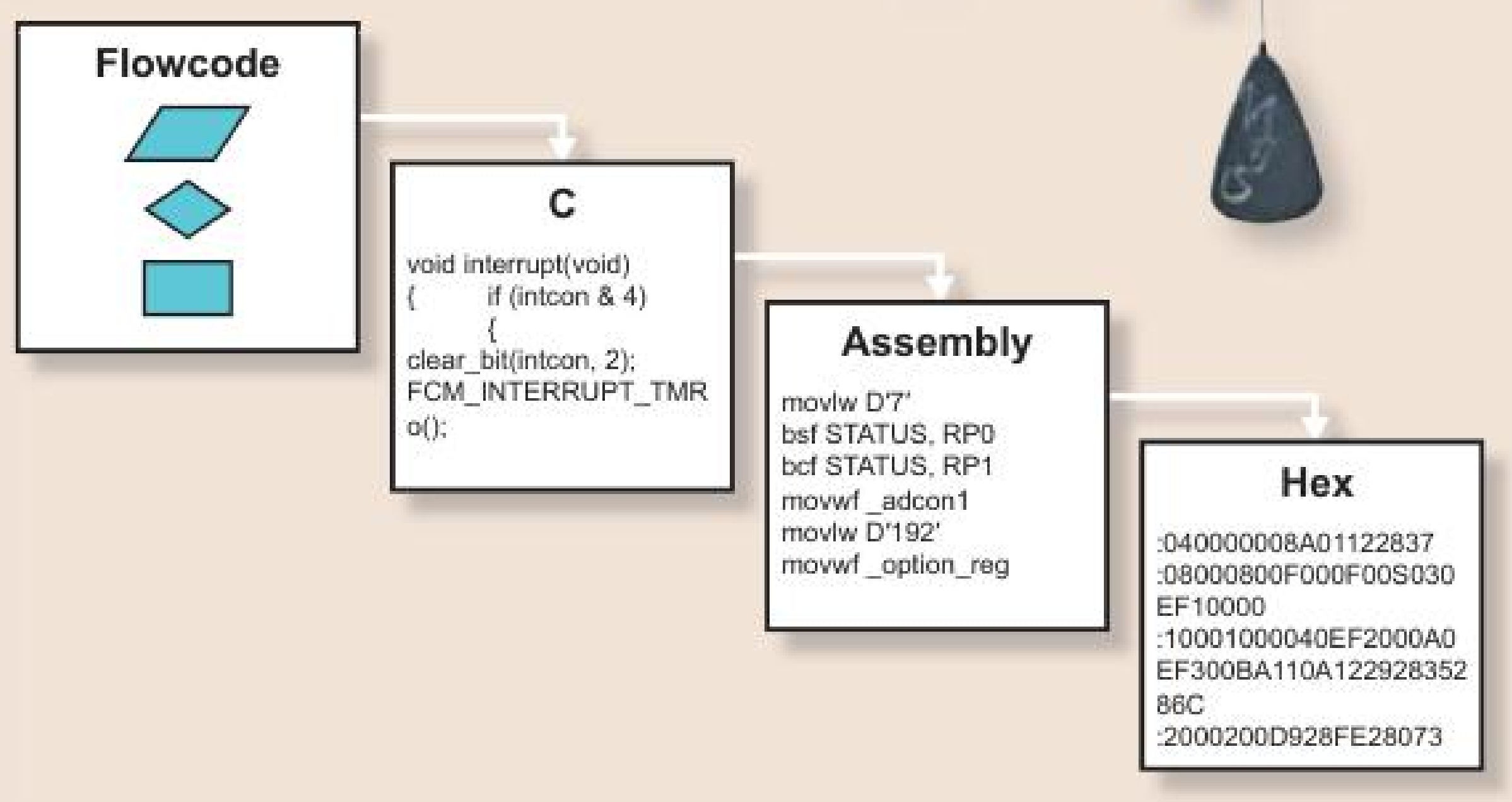
The Microchip name and logo, the Microchip logo, AVR, MPLAB and PIC are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners. © 2021 Microchip Technology Inc. All rights reserved. DS50003138A. MEC2381A-UK-07-21

Practical
Electronics

Volume 51. No. 2
February 2022
ISSN 2632 573X

Contents





```
void interrupt(void)
{         if (intcon & 4)
          {
clear_bit(intcon, 2);
FCM_INTERRUPT_TMR
o();
```

```
movlw D'7'
bsf STATUS, RP0
bcf STATUS, RP1
movwf _adcon1
movlw D'192'
movwf _option_reg
```

```
:040000008A01122837
:08000800F000F00S030
EF10000
:10001000040EF2000A0
EF300BA110A122928352
86C
:2000200D928FE28073
```

Made in the UK.

Written in Britain, Australia, the US and Ireland.

Read everywhere.

© Electron Publishing Limited 2022

Copyright in all drawings, photographs, articles, technical designs, software and intellectual property published in *Practical Electronics* is fully protected, and reproduction or imitation in whole or in part are expressly forbidden.

The March 2022 issue of *Practical Electronics* will be published on Thursday, 3 February 2022 – see page 72.

Projects and Circuits

Arduino-based Adjustable Power Supply by Tim Blythman 14
A handy power supply with voltage and current monitoring and limiting, built as an Arduino shield that's convenient and easy to set up and use.

Battery Monitor Logger by Tim Blythman 22
Knowing the condition of your batteries is essential for keeping them healthy longterm – this project can monitor and log all your battery's vital statistics.

Electronic Wind Chimes by John Clarke 30
Our Electronic Wind Chimes removes your reliance on the wind, and even gives you the possibility of playing your own tunes, enriching the experience!

Mini Digital AC Panel Meters by Jim Rowe 38
We look at low-cost meters designed to measure AC voltages and currents. Some of them can even calculate and display power, energy consumption and frequency.

Series, Features and Columns

Techno Talk by Mark Nelson 8
Should we be worried?

Net Work by Alan Winstanley 10
The latest developments in the electricity generation sector and ideas for streaming favourite radio and music channels – plus a practical 'Net Workshop' topic!

KickStart by Mike Tooley 42
Part 7: Plug and play with I²C

Flowcode Graphical Programming by Martin Whitlock 47
RedBoard and PICs – Part 2

Max's Cool Beans by Max The Magnificent 52
Flashing LEDs and drooling engineers – Part 24

Audio Out by Jake Rothman 56
Analogue Vocoder – Part 4: Driver Amplifier

Circuit Surgery by Ian Bell 60
Op amp offsets – Part 1

Make it with Micromite by Phil Boyce 64
Part 33: Using CRC with iButtons and the Micromite

Regulars and Services

Wireless for the Warrior 2

Subscribe to Practical Electronics and save money 4

NEW! Practical Electronics back issues DOWNLOADS – 2021 now available! 6

Reader services – Editorial and Advertising Departments 7

Editorial 7
Happy New Year!

Exclusive Microchip reader offer 9
Win a Microchip WLR089 Xplained Pro Evaluation Kit

PE Teach-In 8 46

Practical Electronics PCB Service 68
PCBs for Practical Electronics projects

Classified ads and Advertiser index 71

Next month! – highlights of our next issue of Practical Electronics 72

WIRELESS FOR THE WARRIOR

by LOUIS MEULSTEE

THE DEFINITIVE TECHNICAL HISTORY OF RADIO COMMUNICATION EQUIPMENT IN THE BRITISH ARMY

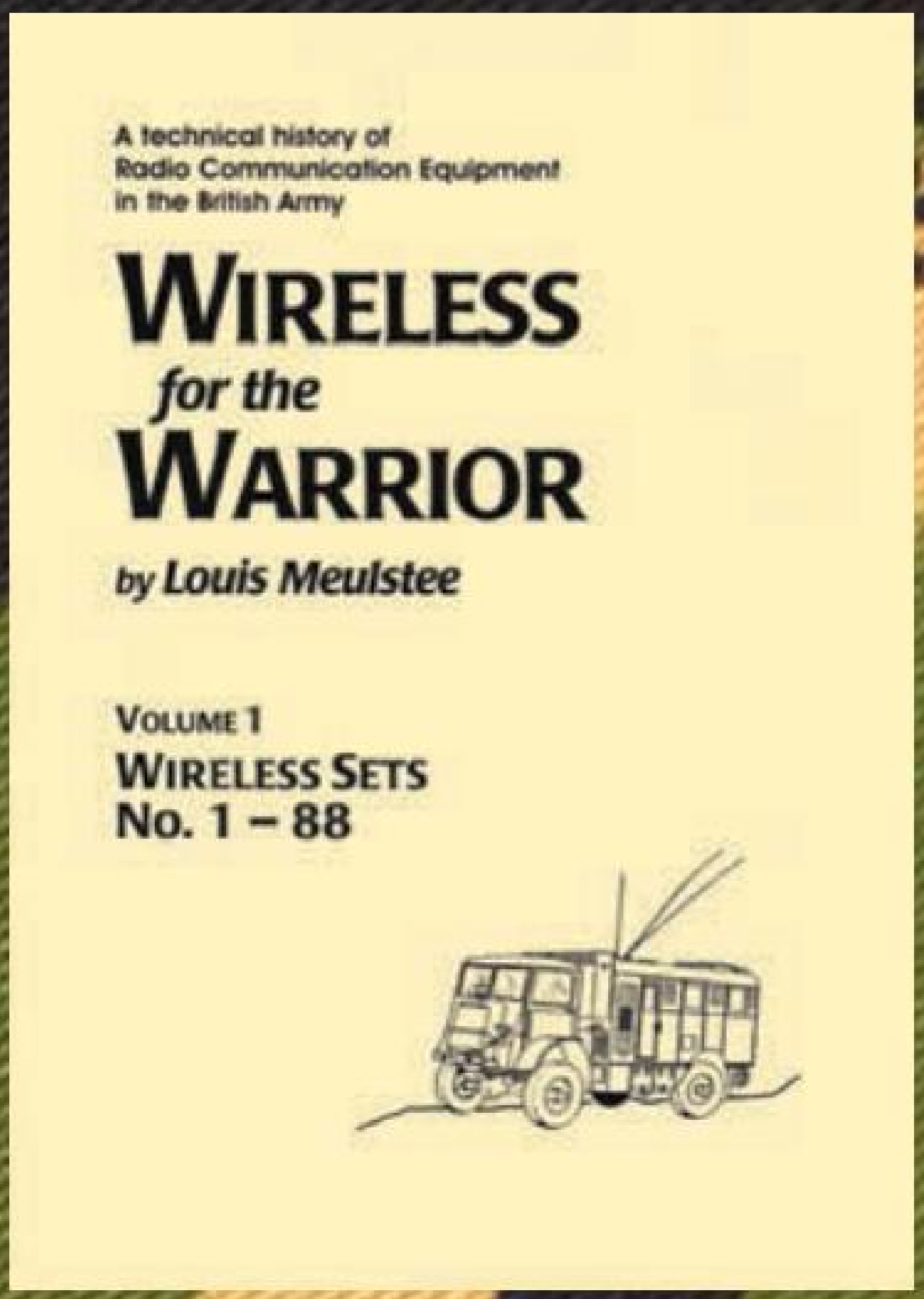
The *Wireless for the Warrior* books are a source of reference for the history and development of radio communication equipment used by the British Army from the very early days of wireless up to the 1960s.

The books are very detailed and include circuit diagrams, technical specifications and alignment data, technical development history, complete station lists and vehicle fitting instructions.

Volume 1 and *Volume 2* cover transmitters and transceivers used between 1932-1948. An era that starts with positive steps taken to formulate and develop a new series of wireless sets that offered great improvements over obsolete World War I pattern equipment. The other end of this timeframe saw the introduction of VHF FM and hermetically sealed equipment.

Volume 3 covers army receivers from 1932 to the late 1960s. The book not only describes receivers specifically designed for the British Army, but also the Royal Navy and RAF. Also covered: special receivers, direction finding receivers, Canadian and Australian Army receivers, commercial receivers adopted by the Army, and Army Welfare broadcast receivers.

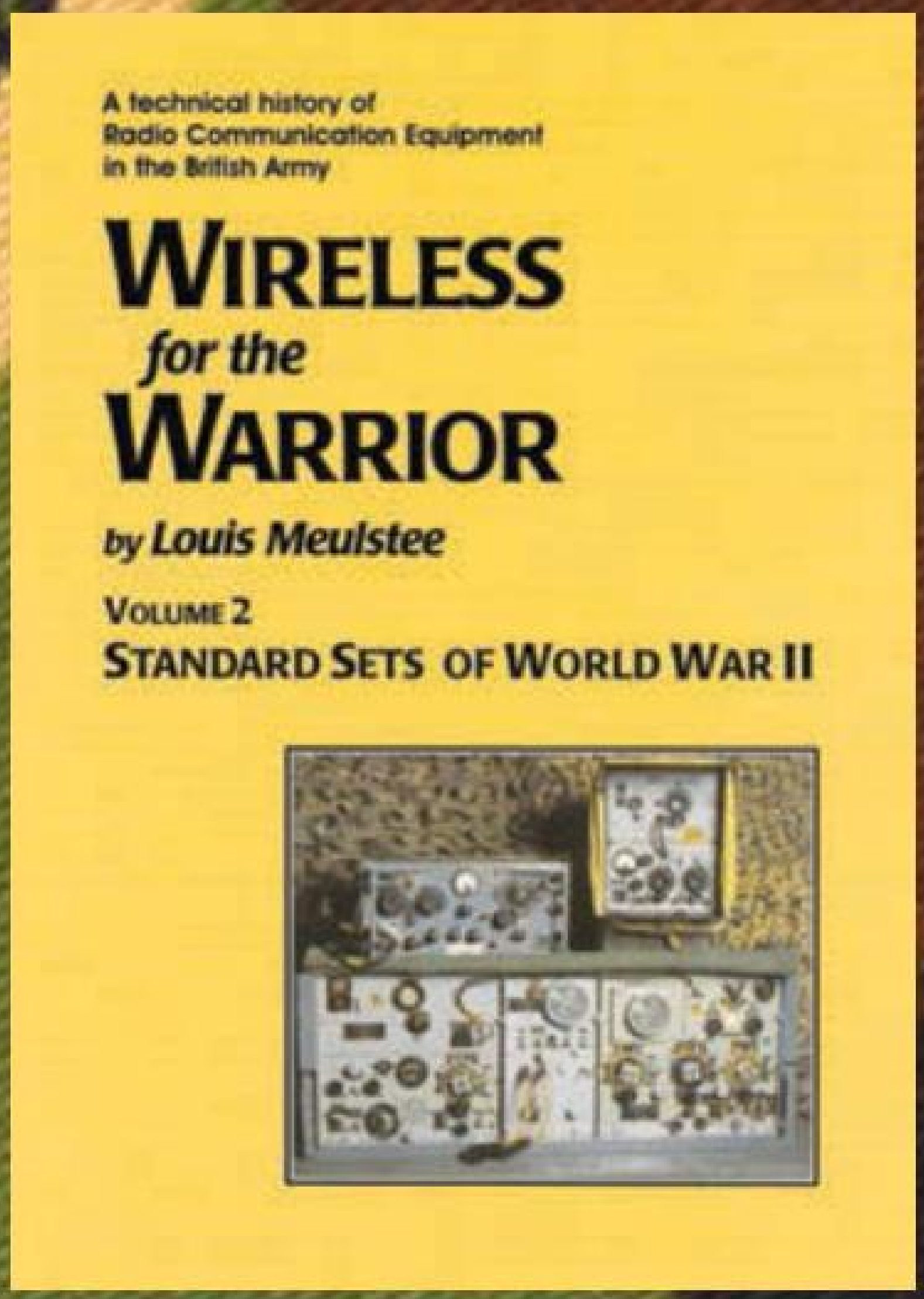
Volume 4 covers clandestine, agent or 'spy' radio equipment, sets which were used by special forces, partisans, resistance, 'stay behind' organisations, Australian Coast Watchers and the diplomatic service. Plus, selected associated power sources, RDF and intercept receivers, bugs and radar beacons.



A technical history of
Radio Communication Equipment
in the British Army

WIRELESS
for the
WARRIOR
by Louis Meulstee

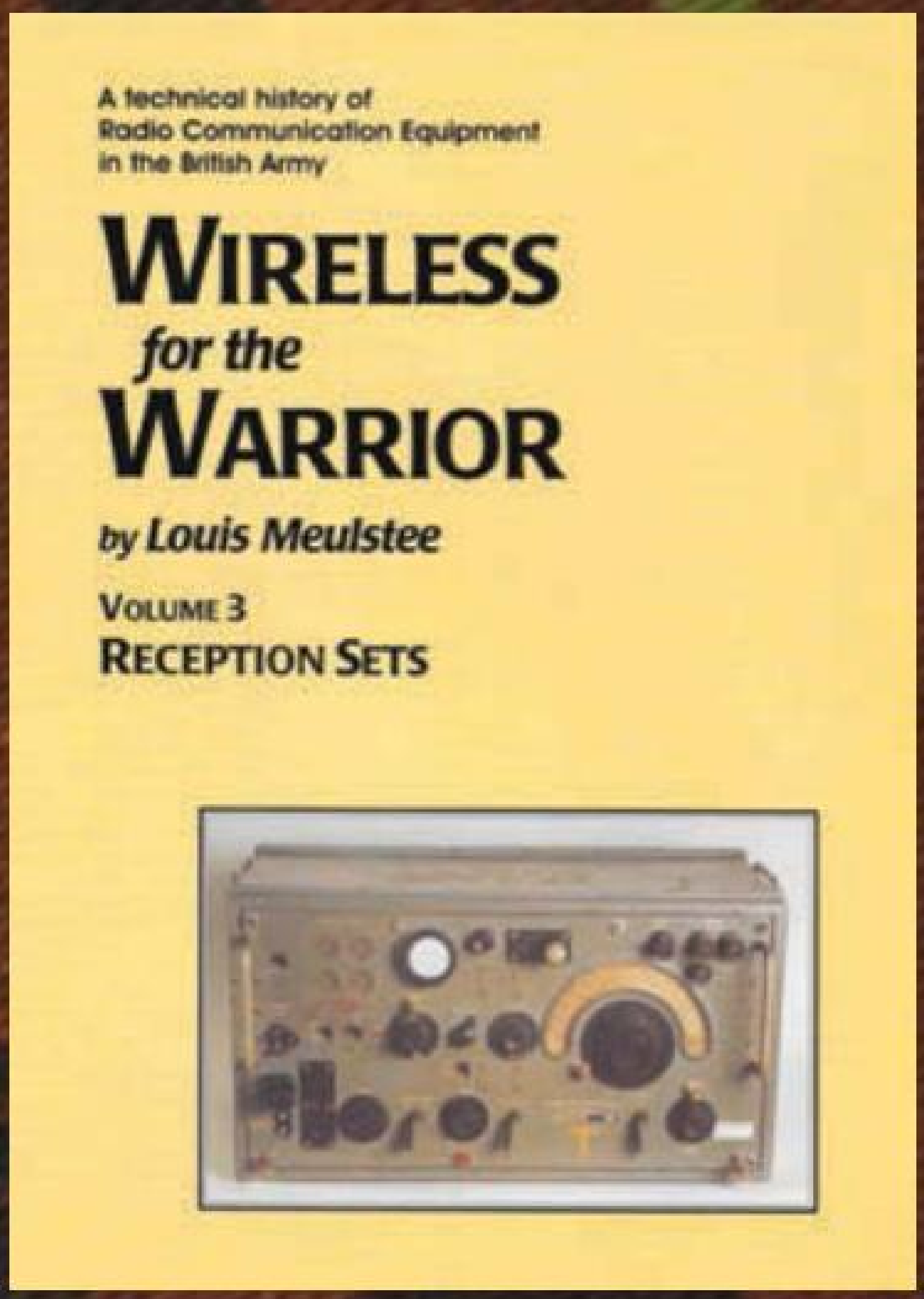
VOLUME 1
WIRELESS SETS
No. 1 – 88



A technical history of
Radio Communication Equipment
in the British Army

WIRELESS
for the
WARRIOR
by Louis Meulstee

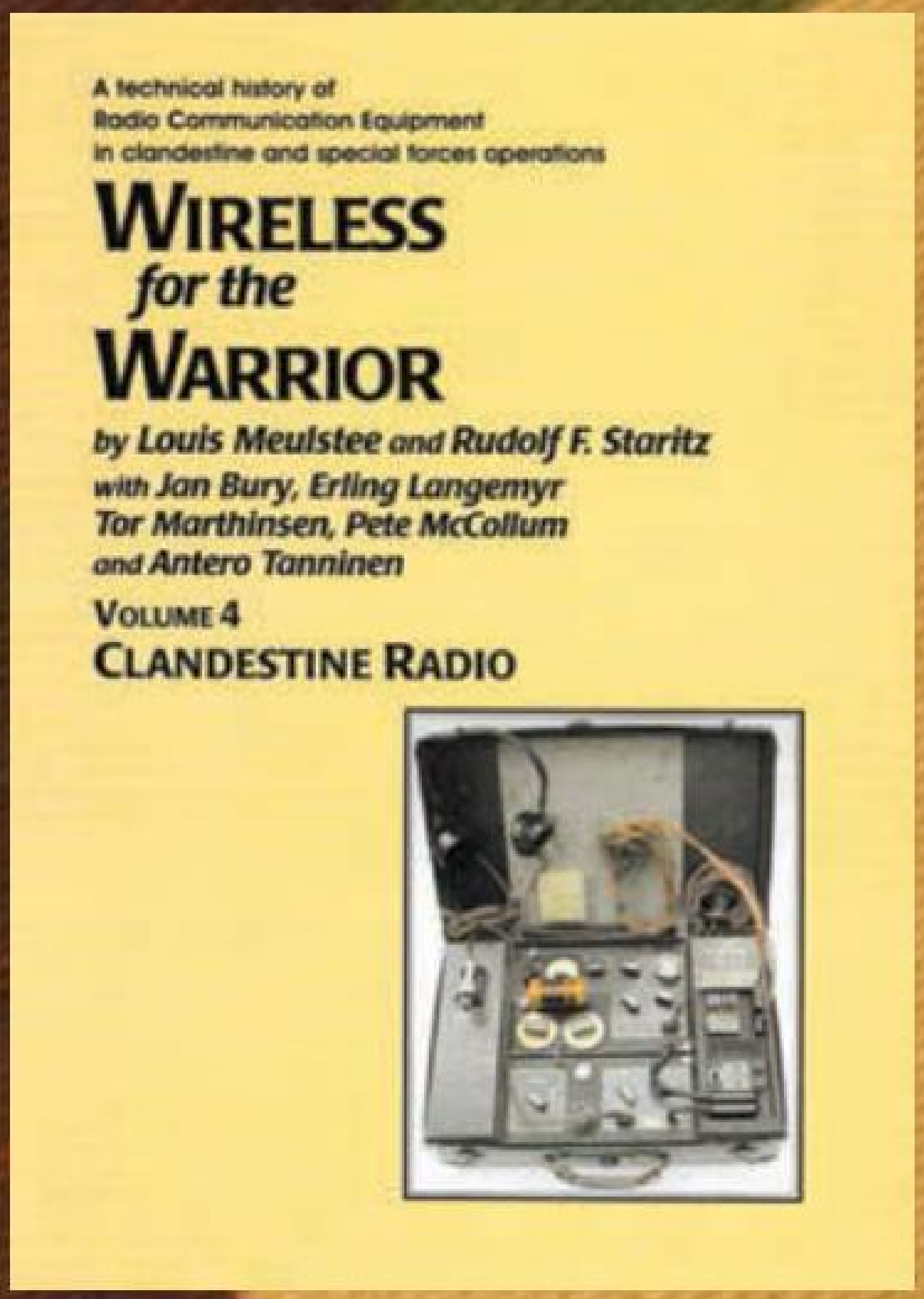
VOLUME 2
STANDARD SETS OF WORLD WAR II



A technical history of
Radio Communication Equipment
in the British Army

WIRELESS
for the
WARRIOR
by Louis Meulstee

VOLUME 3
RECEPTION SETS



A technical history of
Radio Communication Equipment
in clandestine and special forces operations

WIRELESS
for the
WARRIOR
by Louis Meulstee and Rudolf F. Staritz
with Jan Bury, Erling Langemyr
Tor Marthinsen, Pete McCollum
and Antero Tanninen

VOLUME 4
CLANDESTINE RADIO

ORDER YOURS TODAY!
JUST CALL 01202 880299 OR VISIT www.electronpublishing.com

QUASAR
electronics

Quasar Electronics Limited
PO Box 6935, Bishops Stortford
CM23 4WP, United Kingdom

Tel: 01279 467799
E-mail: sales@quasarelectronics.co.uk
Web: quasarelectronics.co.uk

All prices include 20% VAT. Free UK mainland delivery on orders over £60. Postage & Packing Options (Up to 1Kg gross weight): **UK Standard 2-5 Day Delivery -** £4.95 : **UK Mainland Next Day Delivery -** £9.95 : Please order online if you reside outside the UK (our website will calculate postage for you).
Payment: We accept all major credit/debit cards. Make UK cheques/PO's payable to Quasar Electronics Limited and include P&P detailed above.
!! Order online for reduced price postage and fast despatch !!
Please visit our online shop now for full details of over 1000 electronic kits, projects, modules and publications. Discounts for bulk quantities.

QuasarElectronics.co.uk

Solutions for Home, Education & Industry Since 1993

cebek

velleman

Official Main Dealer stocking the full range of Kits, Modules, Robots, Instruments, Tools and much, much more...

SANTA'S Workshop

Electronic Kits & Modules

We have a massive selection of self-assembly electronic kits and pre-assembled modules. Please see the full range on our website or call for details.

LED Buddy / LED Tester Kit

Hold any type of LED to the contact pads to see it's polarity, forward voltage & the recommended series resistor value. Adjustable target current & forward voltage. Great design aid. 1x PP3 battery powered.
*Order Code: MK198 - **£12.73***

5-in-1 Mechanical Coding Robot Kit

Mission-based, entry level STEM (Science - Technology - Engineering - Mathematics) coding robot designed to inspire users to learn about mechanical engineering & coding basics. Solderless assembly. 2x AAA battery powered (not included). Aged 14+.
*Order Code: KSR19 - **£32.95***

Snowman Flashing LED Kit

Have some educational festive fun with this animated LED snowman gadget. 69 multicoloured LEDs (great for soldering practice). Snowflake effect with PWM controlled LEDs. Random generator for a more realistic effect. 1x 9V PP3 battery or 9-12Vdc wall adapter powered. Makes an ideal gift.
*Order Code: MK200 - **£18.58***

Brightdot Clock Kit - BLACK Edition

Brighten any room or space with this fully Arduino® compatible, ESP32 controlled BrightDot clock kit. This designer black edition features 60 bright RGB LEDs that reflect against the surface on which you mount the clock, hence telling you what time of day it is. ESP32 data cable & power supply included.
*Order Code: K2400B - **£117.43***

3 Channel RGB LED Light Organ Kit

3 outputs react to different sound frequencies. On-board microphone picks up surrounding sound or music and drives the low, mid & high frequency outputs. Connect a 12-24Vdc RGB LED strip or separate colour LED strips or LEDs (not included). Master & separate channel sensitivity adjustment. 1.25 A max. per channel. Panel mountable facia 127 x 44mm. Terminal block or jack power supply connection. Control knobs included.
*Order Code: WSL209 - **£14.10***

Stereo Ultrasonic Bat Detector Kit

Converts high frequency sounds (20 - 90kHz) normally imperceptible to humans like bat signals into audible noise. Can also help detect failures in machines, engines, etc. Stereo feature adds the possibility to pinpoint the source. Requires 3xAA batteries. 3.5mm jack output for headphones.
*Order Code: K8118 - **£20.39***

LED Christmas Tree Kit

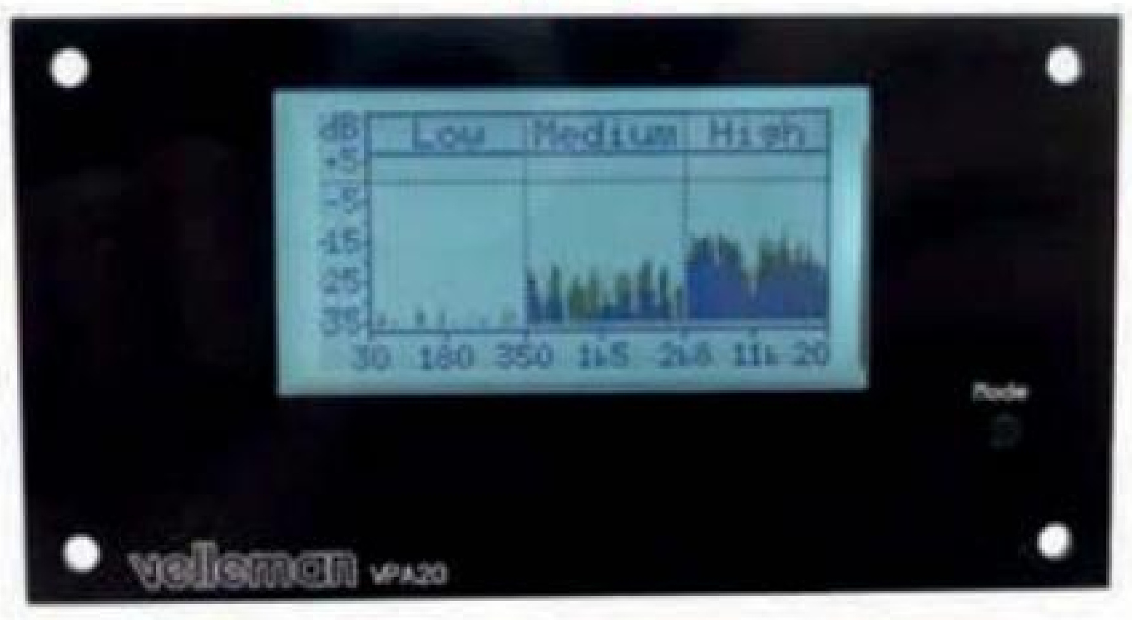
Ho! Ho! Ho! The classic Christmas kit for the budding electronics enthusiast. 15 blinking blue LEDs. Requires 1xPP3 battery. Get cosy…
*Order Code: MK100B - **£7.19***

Card Sales & Enquiries 01279 467799

Digitally Controlled FM Radio Kit

Build your own modern, high quality FM receiver project with excellent sensitivity powered by a simple 9V PP3 battery (not included). Auto-seeking button. 4 station presets. Volume control. Excellent learning project for schools and colleges.
*Order Code: MK194N - **£21.59***

Audio Analyser Display Kit

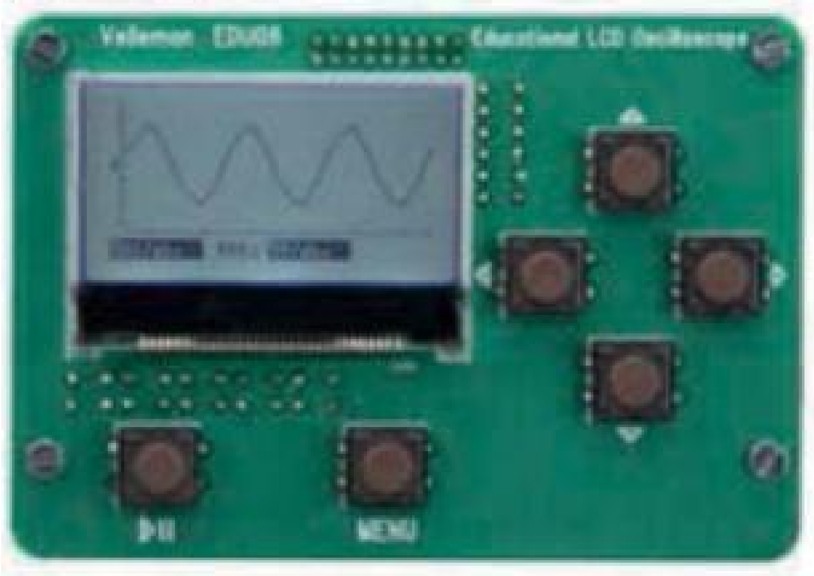


Small, compact LCD display, ideal for panel mounting. Give your homemade audio gear a high-tech look. Upgrade existing equipment. Provides Peak Power, RMS Power, Mean dB, Peak dB, Linear Audio Spectrum And 1/3 Octave Audio Spectrum. Auto / Manual range selection. Peak-hold function. Speaker impedance selection.
*Order Code: K8098 - **£38.39***

Electronic Component Tester Kit

Build your own versatile component tester. Shows value and pin layout information for resistors (0.1 Ohm resolution, max. 50 MOhm), coils (0.01mH - 20H), capacitors (28pF - 100mF), diodes, BJT, JFET, E-IGBT, D-IGBT, E-MOS & D-MOS.
*Order Code: K8115 - **£44.15***

LCD Oscilloscope Educational Kit



Build your own LCD oscilloscope with this exciting new kit. Learn how to read signals. See the electronic signals you learn about displayed on your own LCD oscilloscope. Despite the low cost, this oscilloscope kit has a lot of features found on expensive units like signal markers, frequency, dB, true RMS readouts and more. A powerful auto-setup function will get you going in a flash!
*Order Code: EDU08 - **£48.54***

Practical Electronics

UK readers SAVE £1 on every issue

SUBSCRIBE NOW!

PRICE RISE END OF JAN 22

01202 087631

Take out a one-year subscription and save £10 over the year.
Even better – save £1 per issue if you subscribe for two years – a total saving of £24.

Overseas rates represent exceptional value

You also:
- **Avoid any cover price increase for the duration of your subscription**
- **Get your magazine delivered to your door each month**
- **Guarantee a copy, even if the newsagents sell out**

Order by:
- **Phone or post with a cheque, postal order or credit card**
- **Or order online at: www.electronpublishing.com**

Practical Electronics

SUBSCRIPTION PRICES

Subscriptions for delivery direct to any address in:

UK: 6-months £26.99, 12-months £49.85, 24-months £94.99

Europe Airmail: 6-months £30.99, 12-months £57.99, 24-months £109.99

Rest Of The World Airmail: 6-months £37.99, 12-months £70.99, 24-months £135.99

Cheques (in **£ sterling only**) payable to *Practical Electronics* and sent to: Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

Also via our online shop at**: www.electronpublishing.com**

Subscriptions start with the next available issue. We accept MasterCard or Visa.
(For past issues see the Back Issues page.)

DIGITAL SUBSCRIPTIONS

Electronic subscriptions are available from £24.99 for 12 months, more details at: **www.electronpublishing.com**

SUBSCRIPTION ORDER FORM

☐ **6 Months: UK £26.99, Europe £30.99 (Airmail), Rest of the World £37.99 (Airmail)**
☐ **1 Year: UK £49.85, Europe £57.99 (Airmail), Rest of the World £70.99 (Airmail)**
☐ **2 Years: UK £94.99, Europe £109.99 (Airmail), Rest of the World £135.99 (Airmail)**

To: **Practical Electronics Subscriptions**
PO Box 6337
Bournemouth BH1 9EH
United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

I enclose payment of £ ....
(cheque/PO in £ sterling only)
payable to *Practical Electronics*

☐ Please charge my Visa/Mastercard

My card number is: ....
Please print clearly, and check that you have the number correct

Card Security Code .... Valid From Date........................
(The last 3 digits on or just under the signature strip)

Card Ex. Date ....

Name ....

Address ....

Post code .... Tel. ....

Email ....

Subscriptions start with the next available issue.

Subscribe to SILICON CHIP

SILICON CHIP
DECEMBER 2021
The small but powerful
Hummingbird
Amplifier
Hands-on with the
Raspberry Pi Pico
The smallest Raspberry Pi yet!

Australia's top electronics magazine

Silicon Chip is one of the best DIY electronics magazines in the world. Each month is filled with a variety of projects that you can build yourself, along with features on a wide range of topics from in-depth electronics articles to general tech overviews.

	Print	Combined	Online
6 months	$100 ≈£54 ≈€64	$110 ≈£60 ≈€70	$50 ≈£27 ≈€31
1 year	$195 ≈£105 ≈€124	$215 ≈£116 ≈€137	$95 ≈£51 ≈€60
2 years	$380 ≈£205 ≈€242	$415 ≈£224 ≈€264	$185 ≈£100 ≈€118

- *Combined subscriptions include both the printed magazine and online access.*
- *Prices are in Australian dollars (AUD) unless otherwise specified.*
- *Prices listed are for everywhere in the world except Australia and New Zealand. View siliconchip.com.au/Shop/SubRates for a full list of current prices.*
- *Price estimates are just that and should only be used as a guide, we primarily take payment in the form of Australian dollars.*

Try our Online Subscription – now with PDF downloads!

An online issue is perfect for those who don't want too much clutter around the house and is the same price worldwide. Issues can be viewed online, or downloaded as a PDF.

Published in Silicon Chip

Big Brother is Tracking You; November 2021

USB Cable Tester; November 2021

Hummingbird Amp; December 2021

Raspberry Pi Pico; December 2021

To start your subscription go to siliconchip.com.au/Shop/Subscribe

DOWNLOAD YOUR PE/EPE BACK ISSUES

COMPLETE-YEAR collections – all 12 issues for each year!

From just £6.95 each

VOL 28 – January to December 1999 – **£6.95**
VOL 29 – January to December 2000 – **£6.95**
VOL 30 – January to December 2001 – **£6.95**
VOL 31 – January to December 2002 – **£6.95**
VOL 32 – January to December 2003 – **£6.95**
VOL 33 – January to December 2004 – **£6.95**
VOL 34 – January to December 2005 – **£6.95**
VOL 35 – January to December 2006 – **£6.95**
VOL 36 – January to December 2007 – **£6.95**
VOL 37 – January to December 2008 – **£6.95**
VOL 38 – January to December 2009 – **£8.95**
VOL 39 – January to December 2010 – **£8.95**
VOL 40 – January to December 2011 – **£8.95**
VOL 41 – January to December 2012 – **£13.95**
VOL 42 – January to December 2013 – **£14.95**
VOL 43 – January to December 2014 – **£15.95**
VOL 44 – January to December 2015 – **£15.95**
VOL 45 – January to December 2016 – **£17.95**
VOL 46 – January to December 2017 – **£20.95**
VOL 47 – January to December 2018 – **£25.95**
VOL 48 – January to December 2019 – **£25.95**
VOL 49 – January to December 2020 – **£27.95**
VOL 50 – January to December 2021 – **£29.95 – NEW!**

FIVE-YEAR collections – 60 issues!

From just £14.95 each

PE2832: Jan 99 to Dec 03 – £14.95
PE2933: Jan 00 to Dec 04 – £14.95
PE3034: Jan 01 to Dec 05 – £14.95
PE3135: Jan 02 to Dec 06 – £14.95
PE3236: Jan 03 to Dec 07 – £14.95
PE3337: Jan 04 to Dec 08 – £14.95
PE3438: Jan 05 to Dec 09 – £19.95
PE3539: Jan 06 to Dec 10 – £19.95
PE3640: Jan 07 to Dec 11 – £19.95
PE3741: Jan 08 to Dec 12 – £19.95
PE3842: Jan 09 to Dec 13 – £24.95
PE3943: Jan 10 to Dec 14 – £24.95
PE4044: Jan 11 to Dec 15 – £29.95
PE4145: Jan 12 to Dec 16 – £29.95
PE4246: Jan 13 to Dec 17 – £35.95
PE4347: Jan 14 to Dec 18 – £36.95
PE4448: Jan 15 to Dec 19 – £37.95
PE4549: Jan 16 to Dec 20 – £39.95 – NEW!

20 YEARS OF PE!

AMAZING BUNDLE OFFER
Every issue from 1999 – 2018

240 magazines for just

£55.95

PE2832: Jan 99 to Dec 03
PLUS!
PE3337: Jan 04 to Dec 08
PLUS!
PE3842: Jan 09 to Dec 13
PLUS!
PE3447: Jan 14 to Dec 18

Do you have the 15-year bundle?

Upgrade to the new 20-year bundle for just £9.95 with: PE15to20

Download your collections at: www.electronpublishing.com

Practical Electronics

Volume 51. No. 2
February 2022
ISSN 2632 573X

Editorial offices
Practical Electronics **Tel** 01273 777619
Electron Publishing Limited **Mob** 07973 518682
1 Buckingham Road **Fax** 01202 843233
Brighton **Email** pe@electronpublishing.com
East Sussex BN1 3RA **Web** www.electronpublishing.com

Advertisement offices
Practical Electronics Adverts **Tel** 01273 777619
1 Buckingham Road **Mob** 07973 518682
Brighton **Email** pe@electronpublishing.com
East Sussex BN1 3RA

Editor Matt Pulzer
General Manager Louisa Pulzer
Digital subscriptions Stewart Kearn **Tel** 01202 880299
Online Editor Alan Winstanley
Web Systems Kris Thain
Publisher Matt Pulzer

Print subscriptions
Practical Electronics Subscriptions
PO Box 6337
Bournemouth BH1 9EH **Tel** 01202 087631
United Kingdom **Email** pesubs@selectps.com

Technical enquiries
We regret technical enquiries cannot be answered over the telephone. We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine. We cannot provide data or answer queries on articles or projects that are more than five years old.

Questions about articles or projects should be sent to the editor by email: pe@electronpublishing.com

Projects and circuits
All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it.

A number of projects and circuits published in *Practical Electronics* employ voltages that can be lethal. You should not build, test, modify or renovate any item of mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD (GFCI) adaptor.

Component supplies
We do not supply electronic components or kits for building the projects featured, these can be supplied by advertisers. We advise readers to check that all parts are still available before commencing any project in a back-dated issue.

Advertisements
Although the proprietors and staff of *Practical Electronics* take reasonable precautions to protect the interests of readers by ensuring as far as practicable that advertisements are bona fide, the magazine and its publishers cannot give any undertakings in respect of statements or claims made by advertisers, whether these advertisements are printed as part of the magazine, or in inserts. The Publishers regret that under no circumstances will the magazine accept liability for non-receipt of goods ordered, or for late delivery, or for faults in manufacture.

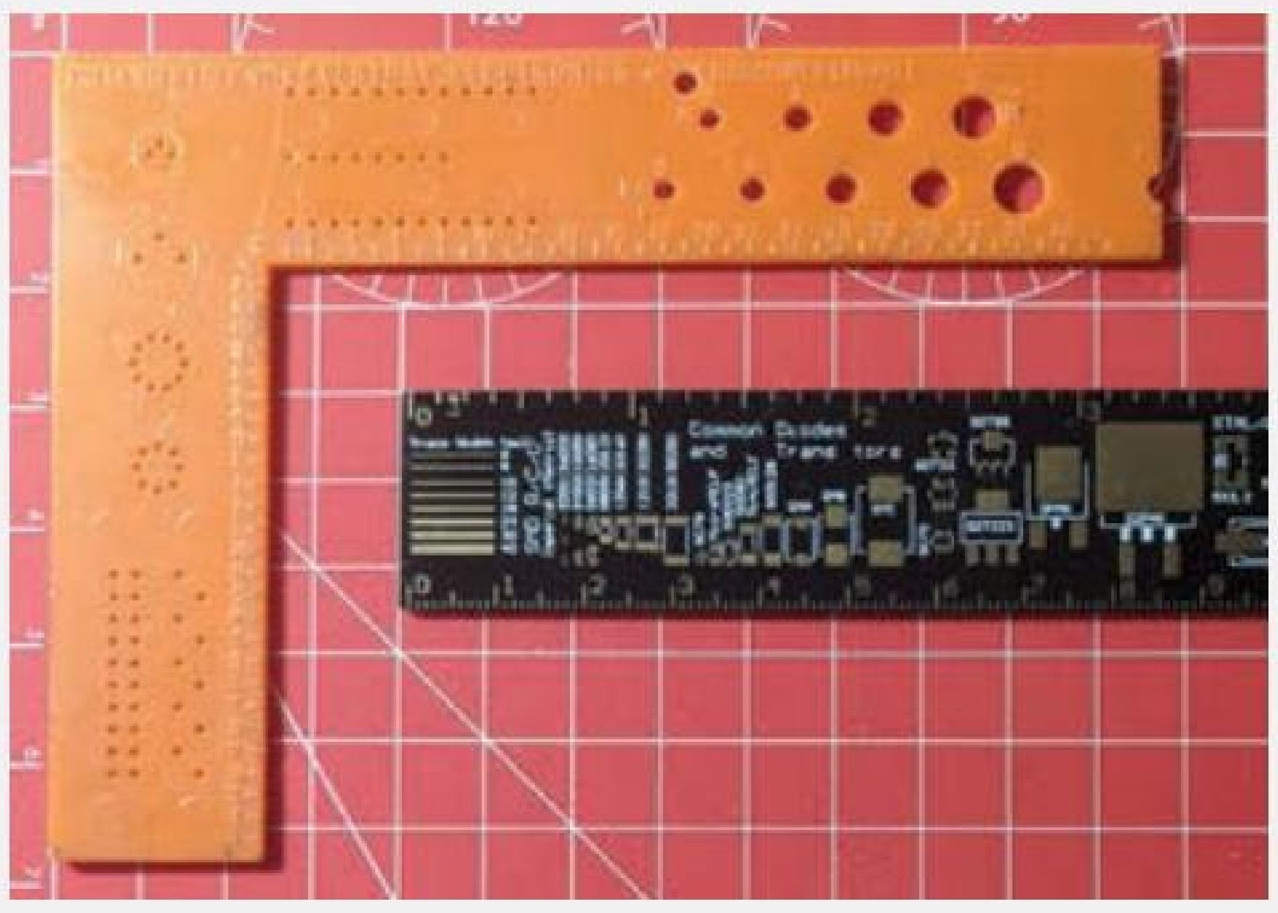
Transmitters/bugs/telephone equipment
We advise readers that certain items of radio transmitting and telephone equipment which may be advertised in our pages cannot be legally used in the UK. Readers should check the law before buying any transmitting or telephone equipment, as a fine, confiscation of equipment and/or imprisonment can result from illegal use or ownership. The laws vary from country to country; readers should check local laws.

Editorial

Happy New Year!

Welcome to 2022 and another year of learning, practising and simply enjoying electronics. We're kicking off the year with some fascinating projects – I particularly enjoyed Tim Blythman's *Arduino-based Adjustable Power Supply*. For many applications it's all you need for a smart, compact PSU.

On the article side, Mike Tooley's introduction to the I^2C bus will get you up and running with this easy-to-use, flexible communication protocol. It lets your microcontroller device or board talk to a wide range of useful modules and sensors using a minimum of connections. Speaking of microcontrollers, *Part 2* of Martin Whitlock's *Flowcode* series demonstrates just how easy it is to program a PIC or RedBoard (a low-cost Arduino clone). The basic version of Flowcode is free, so why not give it a try. I think you'll be impressed with the speed with which you can get an application operating smoothly in hardware *without* resorting to complicated C or assembler programming.



Blast from the past

It's always a pleasure to hear from long-time readers and subscribers. Ian Williams was kind enough to send me an email, saying that he's been receiving, '*EE*/*Practical Electronics* [mostly] on and off for the last 40+ years. I still treasure the PCB/Pin-out ruler I received back then as a junior constructor. I have attached a photo of it with a modern-day equivalent. I wonder how many other 1970s freebies are still around? Love your magazine!' Thank you, Ian, it's always nice to be appreciated – the team that publish *PE* all work very hard to deliver the best magazine we can… but back to Ian's question. The ruler in question was actually an *Everyday Electronics* giveaway (but we count it as one of ours since *EE* eventually joined forces with *PE* to become *EPE* – see Alan Winstanley's fascinating two-part article on our long history: **www.epe-magazine.co.uk/resources.html**) – does anyone out there still have one of the other giveaways that used to accompany the magazine? Do you still use it? We'd love to hear.

Oh dear…

Last, but not least, I'm sorry to report a price rise for *PE*. We are a small business and we've had to face lower revenue during the pandemic, coupled with rising print, paper and distribution costs. We've kept prices the same for over two years, but we will be raising subscription costs at the end of January. If you subscribe/resubscribe now, then the existing price you pay is locked in for up to two years depending on the period you choose. Even if you are only halfway through your current subscription you can renew at the current price and we will simply extend your subscription without interruption, duplication or loss of issues. So, do grab a post-Christmas bargain before the new prices kick in!

Keep well

Matt Pulzer
Publisher

Should we be worried?

Techno Talk

Mark Nelson

Probably not. But you should at least be concerned, which is a different matter, and it certainly makes sense to take a prudent attitude. The subject is of course the global silicon shortage.

If electronics is an educational and rewarding pastime, then it ought to be something that we enjoy doing without gloomy thoughts, Nevertheless, I feel there's a subject that needs to be mentioned, even if it's not desperately cheerful! The trick, I find, is to treat this as a challenge rather than a problem.

Foreboding

Whether you're into electronics as a hobby or a profession (maybe both), you must have read with some foreboding about the growing silicon shortage and wondered how it might affect your hobby – and even disrupt your employer if you work in an industry reliant on semiconductors. Passive components are less at risk, but a slowdown in silicon production has led to a four-fold rise in the price of silicon. This and the effects of Covid-19 on workforces in the chip-making industry, economic conditions in China, a fire in a critical Japanese IC factory and even climate change are all making it unlikely in the short-to-medium term that we'll be able to buy chips and transistors at the low prices that we've all been enjoying over the last decade.

The silicon that's in particularly short supply is the kind of high-end processors used in desktop computers, laptops, gaming consoles and automobiles. The kind of chips used in the projects described in this magazine are less likely to be hit (but it does happen). However, that's not the whole story. Manufacturers may decide it's more profitable to prioritise production towards chips that are intrinsically more lucrative than the 'bread and butter' parts we use. The supply chain may also put the high-value semiconductors at the front of the queue, and when you see the news stories of truly mega snarl-ups at international shipping container ports, you might wonder when on earth these backlogs will be cleared.

Think ahead

The price we'll pay in the future for electronic components is bound to rise, and not only because of the factors just mentioned. Interest rates have been low for several years, but mounting levels of inflation will likely change this. Inland and international distribution costs have already risen rapidly and it's likely that electronic component suppliers will apply price increases across the board, not selectively. Some firms may freeze prices on existing stocks in their warehouses, but I wouldn't bank on it. More worryingly, it may turn out that some manufacturers and/or distributors may decide that it's no longer economic to make/stock parts for which there is declining demand.

This is where things get tricky, so it's time to think strategically and plan ahead. If your next construction product uses any obscure or hard-to-find components, now would be the time to divert your funds into buying as many of these as you think you may need – plus a few more. If (like me) you run a spare-time nano-business producing specialist electronic products for customers, it would be prudent to stock up now rather than later for parts with no obvious alternative – not only for silicon, but also plastic enclosures and optical displays.

Perils of sourcing unobtainium

Unavailability caught me out a couple of years back. The product that I make happens to use what I thought was a very ordinary op amp that would be made for ever… and beyond. Little did I know that it was in fact an 'end-of-life' product and the only UK distributor with stocks was Farnell, who had about 70 of these critters left. Needless to say, I bought the lot. I still look for them occasionally on eBay and recently bought a dozen from a seller in Poland at a ridiculously low price. Usually, however, sellers are asking sky-high prices for them. My most recent order went to a dealer in China, who took my money but never sent the merchandise. I reported this to eBay, who instructed the seller to refund my money. Indeed he did, but only 25% of the amount I had paid. According to eBay, the case is closed because a refund was issued. You win some, you lose some!

There used to a business in London called The Semiconductor Archives, which, despite the somewhat forbidding aspect of their premises (**https://bit.ly/pe-feb22-tt**), had a remarkably eclectic and comprehensive range of ancient transistors with the added bonus of being happy to handle small orders. Firms of this kind still exist in Britain and indeed globally. One that I have used with complete satisfaction is Component Sense in Scotland. On their website you can find the device you need and the number of units in stock (**www.componentsense.com**). However, they are not very keen on selling single items and have a minimum order value.

Steer clear of retreads

The busy road behind my back garden leads to an industrial estate, with plenty of heavy-goods vehicles using it 24/7. It's not just the air pollution that worries me but also the number of lorry tyre treads scattered along the roadside, looking like skins cast off by snakes. Obviously, they are from remoulds, which are evidently not as reliable as the cost-cutting fleet operators imagine they are. Cheap semiconductors are frequently not a bargain either, as this commentary I found on the Web explains (enjoy the English-not-as-first language – to be fair, it undoubtedly puts my Mandarin to shame!).

A lot of this kind of chips are from older PCBs and de-soldered and recreate the surface to make it can be resell on the market. It's a huge industry in Guangdong, China, where you can find a lot of the fake/retread chips. The price of DS3231 retread chips is less than 0.1 to 0.2 USD after been cleaned, plating the legs, and recreate the surface marks, and of course if they can send it to you via Hong Kong, the int-mail price can low down to started from 1 USD. That is why you can find a lot of the sellers who can give you the price like $2, even with the free shipping, from Hong Kong.

Win a Microchip WLR089 Xplained Pro Evaluation Kit

Practical Electronics is offering its readers the chance to win a Microchip WLR089 Xplained Pro Evaluation Kit (EV23M25A) – and even if you don't win, receive a 20%-off voucher, plus free shipping for one of these products.

Microchip's WLR089 Xplained Pro is a hardware platform designed to evaluate the WLR089U0 LoRa modules.

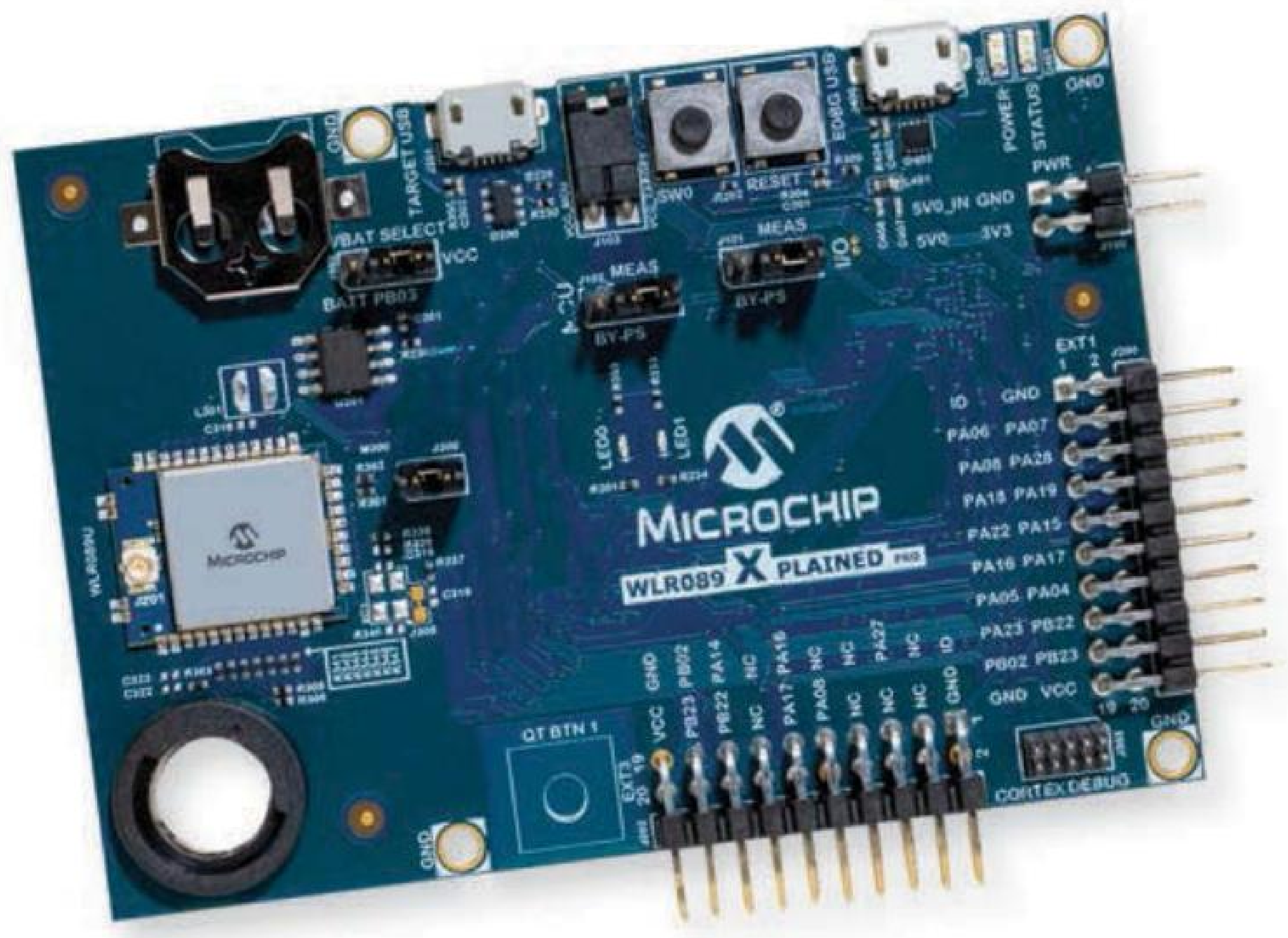
Worth $113.99 (approx £85.00) each

The WLR089U0 module is an ultra-low-power, regulatory-certified LoRa module based on the ATSAMR34J18 LoRa IC. The SAM R34/R35 is a highly integrated family of LoRa technology devices that includes an ultra-low-power, high-performance 32-bit microcontroller (MCU), a LoRa transceiver and a software stack. With their certified reference designs and proven interoperability with major LoRaWAN gateway and network providers, the SAM R34/35 devices significantly reduce time to market for Internet of Things (IoT) end devices.

The WLR089U0 standalone module includes a 32-bit ARM Cortex-M0+ processor and offers 256KB of Flash and 40KB of SRAM (8KB battery backed) in a compact 17x13.5mm package. With ultra-low-power sleep currents as low as 790nA, the WLR089U0 modules are ideal for battery-powered remote sensor applications. The WLR089U0 modules are FCC, IC and RED certified. The WLR089 kit is not only an evaluation platform but also an excellent reference design for developing SAMR34/35-based LoRa end-node applications.

This kit is supported by the Atmel Studio, an integrated development platform, which provides predefined application examples. The kit also provides easy access to various features of the WLR089U0 module and ATSAMR34J18B device, and offers additional peripherals to extend the features of the board and ease the development of custom designs.

Free-to-enter competition



Microchip WLR089 Xplained Pro Evaluation Kit

How to enter

For your chance to win a Microchip WLR089 Xplained Pro Evaluation Kit or receive a 20%-off voucher, including free shipping, enter your details in the online entry form at:

https://page.microchip.com/PE-WLR089.html

Closing date

The closing date for this offer is 31 January 2022.

October 2021 winner

David Widdup

He won a
Microchip PIC32
WFI32E Curiosity Board

Net Work

Alan Winstanley

This month's *Net Work* looks at the latest developments in the electricity generation sector and ideas for streaming favourite radio and music channels. But first, a practical 'Net Workshop' topic!

Half a century ago, when the author took his first steps in constructing amateur electronic projects, most designs were built on stripboard and housed in aluminium or plastic Veroboxes, or universal BIM boxes (the 1970s answer to Australia's Jiffy boxes). In the writer's case, a meagre assortment of workshop tools centred round a hacksaw, a few files and a hand-cranked drill for preparing panels for switches or controls. A teenager could manage it, but any tough steelwork, such as drilling or punching large holes for panel meters in a steel chassis meant calling for Dad and his electric drill.

All the tools you need

These days, hobbyists are totally spoilt for choice, with a dazzling variety of power tools and accessories that empower home constructors like never before. Since the 1980s, cordless drills and drivers have caught on, although it was soon discovered that rechargeable Ni-Cad batteries had a short lifespan with a poor 'memory effect' and seldom were they ready for action when needed. Worse, if spare batteries weren't available then usually the whole tool had to be scrapped once the battery failed.

Tired of wasting batteries and throwing away expensive tools, in the mid-2000s I invested in Ryobi's OnePlus system which uses an 18V Ni-MH battery to power a range of 'bare' tools. It was a forward-thinking idea that would prove to be well ahead of its time. The product's major USP was that batteries were exchangeable, and one battery fitted every Ryobi OnePlus tool: no need to invest in multiple battery packs. Initially, barely a dozen such tools were available, but Ryobi now offers more than a hundred in Europe, catering for a wide range of domestic tasks. Their cordless drills and drivers will appeal to DIY users and hobbyists but there are Ryobi OnePlus tools to help tackle many other home and leisure tasks: mowing, trimming, cleaning, power washing, vacuuming, grinding, sawing, spraying, stapling, tyre inflating, illuminating, cooling and more.

Ryobi's OnePlus 18V tool system includes a soldering iron, rotary tool and USB powerbank adaptor. Note that 18V batteries and chargers aren't included with 'bare tools'. Third-party battery and power take-off adapters are sold separately.

Some readers might be interested in the Ryobi ONE+ soldering iron R18SOI-0, a self-contained iron that clips onto their 18V OnePlus battery and would be good enough for quick repairs or working in the field. A mini rotary tool (R18RT-0) has a flexible drive shaft for hobby projects, or there's the new RRT18-0 pencil-style rotary tool. Ryobi's mini 18V air compressor R18AC-0 might suit bench-top use, but single-phase oil-free compressors can cost much less. There's even a USB accessory that turns a Ryobi 18V battery into a high capacity 5V powerbank, the R18USB-0. This and a Ryobi lantern are on standby here at home (along with an always-on battery charger) during the current season of stormy weather and power cuts. A variety of Ni-MH battery capacities is available from Ryobi – 'starter kits' often include a tool and battery or two bundled with a mains charger, and are a good place to start.

All such tools have limitations, and sometimes you just can't beat the clout of a mains-powered drill. One old ONE+ power drill/driver expired with a 'phut' when trying to screw tannalised timber sections together, but all was not lost as the battery was re-deployed on a replacement drill. In fairness, not every implement quite hits the spot, as some disappointed owners of Ryobi 18V lawnmowers have found, so it's worth checking reviews first. Details of Ryobi's European range will be found at, **https://uk.ryobitools.eu**

For readers in the US, there's a much larger market – view the nearly 200(!) attachments at: **www.ryobitools.com**

Build a cordless toolkit

If you are just starting out in DIY or hobby electronics, it's worth considering gradually building up an 18V toolkit as the expanding Ryobi OnePlus system has withstood the test of time – check Amazon and use the Cameliser plugin in your web browser to alert about price drops. Alternatively, Bosch offers its '18V POWER FOR ALL' system while Hitachi, DeWalt, Makita, Festool and others have pricey 18V and 40V systems for trade users.

More recently, unofficial adaptors made by third parties now allow other brands of 18V battery to be fitted to Ryobi tools. So-called 'power wheel' adapters also allow batteries to be harnessed for kids' motorised cars or bicycles, skateboards, portable lights, signs and so on, and hobbyists might be interested in using these wire-ended adaptors to power electronic projects or, say, LED effects. Search eBay and Amazon for them but, given the high capacity of these Ni-MH batteries, check cable ratings closely and do add an inline fuse.

Streaming sensations

Another ONE+ accessory that caught my eye was a new Bluetooth loudspeaker (Ryobi PAD02B) sold in the US by Home Depot (America's answer to B&Q) but only just arriving in Europe. **Amazon.com** lists the Ryobi Bluetooth speaker at $29.00 + $10.00 shipping and duty to the UK, say £30. (The UK Ryobi website lists it at £51.99). It would probably be ideal for kicking around on the workbench, but it also reflects the trend towards listening to streaming audio via Wi-Fi or a mobile phone, which is the subject of my next topic.

For home network users, gone are the days when radio listeners had to rely on receiving a strong FM or DAB signal, with all the aerial jiggling and

The JBL GO 2 is a compact but punchy waterproof Bluetooth speaker for streaming audio wirelessly.

tuning that it entailed. Some recent 'retail therapy' saw the delivery of a Bluetooth loudspeaker made by JBL. The JBL GO 2 is a very compact waterproof (IPX7) speaker that enables streaming audio to be enjoyed in, say, the shower or garden. It has a surprisingly rich tone for such a small unit – not 'tinny' at all. Available in several colours, the rechargeable speaker can last several hours between charges, and it paired fuss-free with an Android phone. A separate 3.5mm Aux In jack is fitted for times when Bluetooth isn't available, and it can double as a speakerphone for hands-free conference calls or for quizzing Google Assistant out loud (both of which depend on using your smartphone's microphone). The JBL GO 2 was remarkable value for a street price of under £20. Newer models (the pricier GO 3) are available, and there is, of course, a plethora of Bluetooth speakers sold by many other brands. Some also have a MicroSD slot to play audio files that way.

Bluetooth is also built into many portable and tabletop radio receivers, TV sound bars, hi-fi units and so on, which enables audio to be streamed wirelessly to them from, say, a smartphone, laptop or tablet. Some devices incorporate NFC (Near Field Communications) which makes pairing as simple as touch-tapping it with a compatible smartphone: look for the NFC logo on the case.

Where Bluetooth isn't included in audio equipment then some easy options are available. Inexpensive Bluetooth *transmitters* are sold on eBay that plug into a 3.5mm headphone jack, allowing audio to be streamed, range permitting, to Bluetooth headphones, a sound bar or earbuds. Similar Bluetooth *receivers* are sold that fit an audio system's Aux In socket (phono to 3.5mm adaptors are widely available), but versatile Bluetooth Tx/ Rx *transceivers* will work in either mode. These adaptors are rechargeable or might draw power from a USB port, but they can sometimes be frustrating to set up.

This SONRU Bluetooth 5.0 Transceiver has both a TOSLINK optical and a 3.5mm Aux jack. Select Tx or Rx mode as needed. (amazon.co.uk)

Secret source

Next, what about a streaming audio source? Spotify Free offers unlimited songs sprinkled with adverts, or a paid-for service costing £9.99 per month. Amazon Prime membership bundles a selection of two million songs to enjoy on a PC or Fire TV Stick, which is also searchable using Alexa voice commands. Access to Amazon's full catalogue of 75 million songs costs from £7.99 a month. Apple Music sells a variety of plans with different monthly tariffs as well. There's more than just music available online: Amazon's Audible package offers podcasts and audio books for £7.99 a month, while **https://audiobooks.co.uk** (or **.com**) has a free 30-day trial then £7.99 for two books a month. Audio magazines and meditation or natural soundtracks might be available as well.

As alternatives to DAB or FM radio, a myriad of radio stations stream their programmes online, accessible through apps or websites such as the free radio player (go to: **www.onlineradiobox.com**). Sometimes, your scribe will be found outside, tuned into Spain's *Spectrum FM Costa Blanca* streaming over Wi-Fi from the router indoors, with earbuds offering a convenient hands-free listening solution when paired to a smartphone. This system works well, and the range of the author's domestic Wi-Fi has proved surprising. (A mobile app such as WiFi Analyzer displays any SSIDs in range and their signal strength.)

Back indoors, another option is to stream Internet radio channels onto a suitable radio receiver such as the PURE Evoke or the much-missed PURE Avanti Flow (which is exactly what the writer is doing as he types this month's copy). Radios usually need Wi-Fi, although some have Ethernet ports as well. A few pricey streaming radio sets are sold under the classic Roberts name and countless other Internet radios are sold under less familiar brands.

Whether to migrate one's listening pleasures entirely to downloads is a matter of personal taste. Most music is now streamed online, but many of us still enjoy the tactile experience of playing CDs. (It's also worth noting that vinyl record and cassette tape sales are resurging.)

Keen audio enthusiasts might like the Brennan B2, a self-contained CD ripper and amp which has a built-in Raspberry Pi module and will rip and store thousands of CDs. A 2TB disk can host up to 9,000 CDs as MP3s or 4,400 in lossless FLAC format. The Brennan B2 will also stream Internet radio stations, rip vinyl records to MP3, back up to external drives or appear on a LAN as a NAS music drive. Dongles are needed for Wi-Fi and Bluetooth. Brennan also claims to be the first such device to pair with a wide number of Sonos speakers. For network media users, Brennan states DLNA can be installed separately via a software patch (check for details). The Brennan B2 2TB model lists at £569 and readers can check the specs at: **www.brennan.co.uk**

The winds of change

In last month's issue I outlined some bold plans by Rolls-Royce Group to manufacture small modular reactors (SMRs), which are nuclear-fuelled power stations that are 'assembled' onsite from separate factory-built units. More countries are looking for 'clean fuel' SMRs as an answer to the energy crunch, especially as fossil fuel supplies are becoming more restricted, and geopolitical tensions between Germany, Ukraine and gas-supplier Russia are creating a great deal of uncertainty.

It's early days for SMRs, but Rolls-Royce has ambitions to produce them at a rate of up to ten power plants a year based on small, pressurised water reactors, and has submitted plans to regulators for its 470MW design, according to *World Nuclear News*. Two sites

Brennan B2
BACK
NEXT
EJECT
PLAY STOP

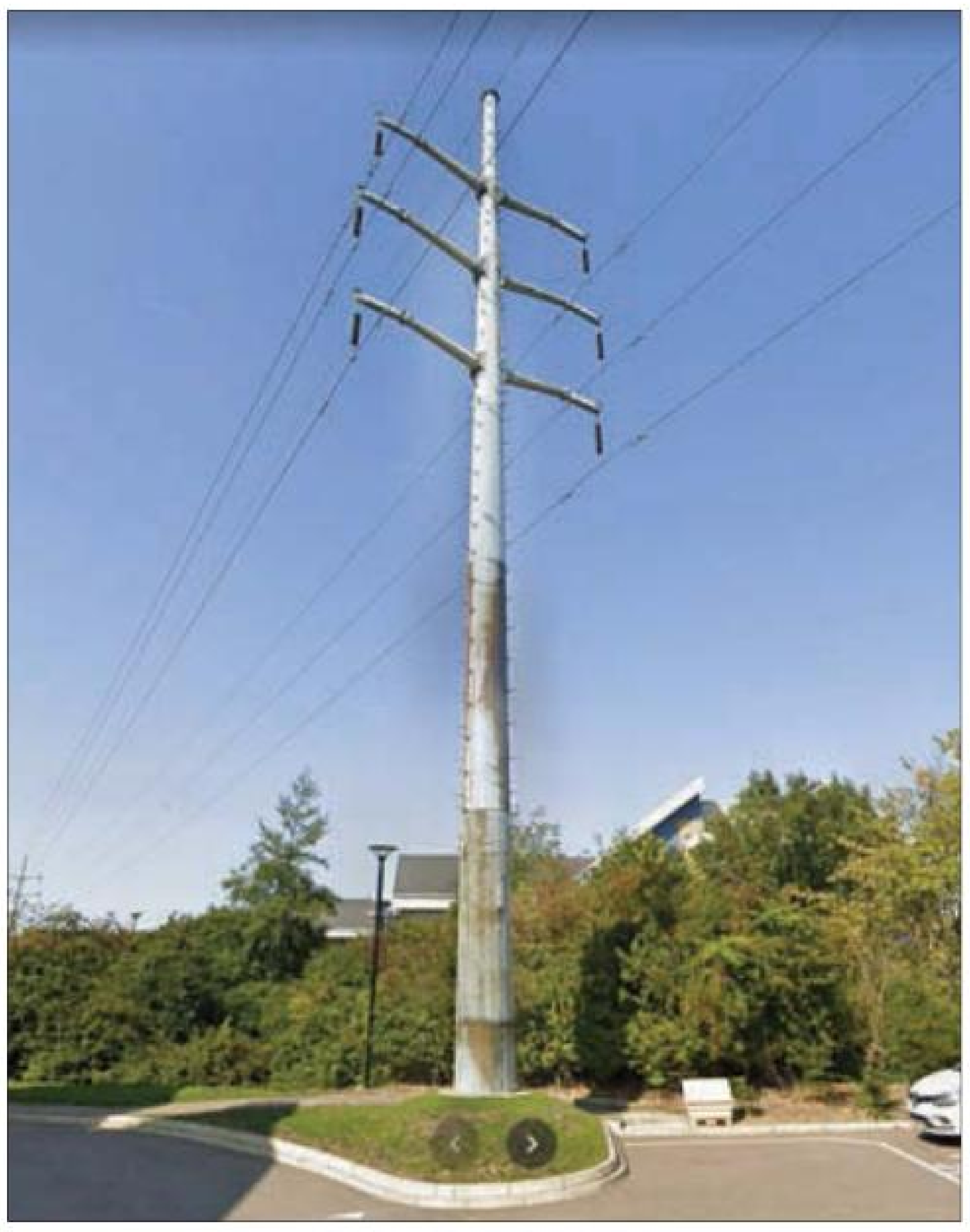
For audio enthusiasts, the Brennan B2 plays and rips CDs onto built-in storage, plays Internet radio and drives Sonos speakers directly. Check the website closely for the full specifications.

in north Wales have been earmarked as possible locations for Rolls-Royce SMRs, and a *Memorandum of Understanding* was recently signed with Sheffield Forgemasters to fabricate precision castings for them.

Rolls-Royce faces stiff competition in the global SMR market. America's NuScale Power is backed by multinational engineering giant Fluor, and has already gained approval for its own scalable SMR power plants in the US, claiming to be the first company in America to do so following lengthy pilot trials dating back as far as 2003. NuScale's 77MW 'VOYGR' power modules will be factory-built and can be shipped by road for final assembly onsite. Multiple modules can be strapped together to build larger capacity: a VOYGR-12 will generate over 900MW of electricity, they say. The first completed VOYGR plant is expected by the end of the decade, and NuScale has also been an eager participant in the UK's SMR plans since 2013.

Possibly with current geopolitics in mind, the US Department of Energy is collaborating with the Ukrainian government to fund a 'safety analysis report' that would be 'available to any utility [company] in Ukraine pursuing US SMR technology'. This would undoubtedly pave the way for the adoption of American SMR power plant in Ukraine; Romania is also signing up to a 6-module NuScale SMR, following a feasibility grant paid by the US Trade and Development Agency, says *World Nuclear News*. Two years earlier, GE Hitachi Nuclear Energy announced collaboration with Poland to manufacture a cost-effective 300MW SMR for the Polish market. More recently, the same SMR was ordered by Ontario Power Generation for completion by the end of the decade as Canada moves towards a fossil-free 'clean fuel' future. There's more: America's Holtec International has teamed up with South Korea's Hyundai to produce a 160MW SMR for installation in the US and then worldwide. The direction of travel away from fossil fuels is clear, but it will likely be the end of the decade before SMRs come on stream, doubtless leaving existing gas and electricity supplies under greater pressure until then.

The roll-out of Britain's wind power programme continues as GE signed off last month the 'financials' for the third tranche (Phase 'C') of the Dogger Bank Wind Farm sited off the east coast of England. They claim Dogger Bank will be the world's largest offshore wind farm, and Phase 'C' will use 87 uprated 14MW Haliade-X wind turbines, the largest currently available, making 277 turbines in total. The UK Government is also investing another £20m in tidal stream projects, with some developments aimed at the Welsh coastline. Unlike wind power, tidal energy is predictable and consistent. UK Government reports state that £285m is being invested annually in this form of marine technology across the UK.



T-shaped pylons in Cambridge Science Park. (Image: Google Street View)

Another Model T (pylon)

In last December's issue I mentioned a new T-shaped design of electricity pylon that is putting in a limited appearance in the UK. My thanks go to reader Andrew Parker of Cambridge, who writes: 'I really enjoy reading your *Net Work* column every month, and I was particularly interested in your December article about the first new pylon design in Britain for more than 100 years. While they may claim to be the first redesign to be deployed in quantity, these new T-pylons in Somerset, England aren't radically different to similar single-pole pylons erected nearly 40 years earlier, when the Cambridge Science Park was extended in the 1980s.

'To enhance the site, a more elegant design was used for the short section where the electricity cables passed through the Science Park. Quite an impressive feat, and presumably very

HAMMOND
new!
1455F extruded flanged enclosures
Learn more: hammfg.com/1455f
Contact us to request a free evaluation sample.
uksales@hammfg.com • 01256 812812

costly, showing the power and influence of Trinity College – the landlord! I think they were replacements for the existing conventional pylons. These four Cambridge pylons can still be seen on the latest Google Street Views. They need a coat of paint! A picture of one of them is on the first page of a website dedicated entirely to photos of UK pylons, of all things, at: **www.gorge.org/pylons/envision**.' Andrew kindly sent a screenshot. The pylons are quite striking and it's a shame that they are being allowed to decay.

Good news for the UK's successful satellite industry – OneWeb is aiming to relocate production to the UK, in partnership with Airbus.

Other news

At the time of writing, all eyes are on French Guiana as we await the launch of the James Webb Space Telescope, currently being installed on board an Ariane rocket for launch on 22 December. The space observatory is a joint venture between NASA, Europe's ESA and Canada's Space Agency, and it carries a 21-foot mirror made of 18 gold-plated hexagonal segments. It will gradually supplant Hubble as the West's key space observatory. More information about the instruments it carries can be found at: **https://bit.ly/pe-feb22-obs** and space fans can see details of various JWST deployments at: **https://bit.ly/pe-feb22-webb**

Satcomms firm OneWeb is aiming to relocate its satellite production from Florida to the UK (see *Net Work*, May 2019). Partnered with manufacturer Airbus, OneWeb's second-generation satellites are expected to roll out from the middle of the decade. Satellites are mostly launched from cosmodromes in Russia or Kazakhstan.

Rolls-Royce has made what it claims is the world's fastest flight in an all-electric plane (dubbed The Spirit of Innovation – see December issue). It's emerged that the airframe was an off-the-shelf kit – the Nemesis NXT racing plane, which Rolls-Royce chose, sensibly enough, as a testbed for its electric propulsion technology. More at: **www.nemesisnxt.com/kit/index.php**

In the UK, the dilatory roll-out of electric vehicle charging points lags well behind the uptake of EVs among consumers. Britain's Society of Motor Manufacturers (SMMT) says one in five new cars sold in November was an EV, and in fact I can't remember when I last saw a TV advert for a petrol or diesel car. New British homes, supermarkets and workplaces, as well as any undergoing major renovation, will be forced to install EV vehicle charging points from 2022, forthcoming Government regulations will state. That is all well and good for urbanites living in new homes or shopping in new supermarkets, but those of us living in rural areas – places where a diesel 4x4 is often essential transport – face great practical challenges. I doubt EVs will fare well in typical wintry storms or countryside snow or mud. The National Farmers Union (NFU) hopes to work with InstaVolt to build a network of charging points on private land or at farm shops. Judging by the PR, the deal struck with NFU Energy seems to be as much about helping farmers and landowners cash in by offering a charging service, as it is about encouraging rural motorists to somehow adopt EVs.

Car breakdown service Britannia Rescue says that 37% of callouts for EVs were due to wheel and tyre problems – blaming the [up to] 50% extra weight EVs carry due to their battery packs. This is twice the call-out rate of petrol/diesel cars having wheel problems. Interestingly, EVs were also three times more likely to break down because of wheel and tyre problems rather than their batteries going flat, Britannia Rescue said.

Planet of the Apes

Elon Musk, of SpaceX, Tesla and PayPal fame, is not only into space-based broadband or electric cars: in between times his Neuralink technology company is addressing the thorny topic of interfacing technology directly with the brain. I struggle for words when it comes to describing the implantation of chips directly into one's brain, so I will offer the Neuralink YouTube video of a cheerful monkey happily playing Pong at: **https://youtu.be/rsCul1sp4hQ** instead. Neuralink hopes its technology will initially help those with spinal cord injuries.

Amazon Fresh has expanded its UK bricks-and-mortar real estate with two more 'Just Walk Out' stores in London, making ten in all (see *Net Work*, May 2021). UK supermarket giant Sainsbury's has also opened its own 'Smartshop Pick & Go' till-free store in Holborn Circus, London. The store is Amazon's first partner internationally to utilise Amazon's own Just Walk Out systems. The Smartshop app is needed on a smartphone, a QR code is scanned on entry and then payments are charged to the linked payment card on exit. A 'till receipt' is emailed afterwards.

Finally, this month, onto the *Net Work* naughty step goes Google, after the author's Home Hub (now called Nest Hub) LCD display locked up without warning. Readers will know that I opted for Google Mini smart speakers (I have three dotted around) plus a desktop Home Hub LCD screen, but several months ago the LCD screen suddenly froze solid. Many hours were wasted rebooting, reinstalling and reconfiguring it on the Google Home app, all to no avail. Online forums highlight other Home Hub users having the same problem, with lock-ups of these first-generation screens being blamed on a firmware or system upgrade gone wrong.

It's unacceptable for new-ish and perfectly workable products to be 'bricked' this way through no fault of the owner. Attempts to reach technical support go unanswered, and Google's PR has yet to respond after I contacted them. It's no surprise that the global total of electronic waste has reached 57 million tonnes this year, according to the WEEE Forum, and it looks like my perfectly good bit of hardware will join it. Instead of buying another, I'll use Google Chrome on a PC and my webcam's microphone.

Last, a reminder that all links will be ready-made for you in the blog on our website, so do check: **www.electronpublishing.com**

Join me next month for another tasty serving of *Net Work!*

The author can be reached at:
alan@epemag.net

Arduino-based Adjustable Power Supply

By
Tim Blythman

We have published all sorts of fancy bench supplies over the years: linear, switchmode, hybrid, high-voltage, high-current, dual-tracking… But sometimes, all you need is a basic power supply with voltage and current monitoring and limiting; something that's convenient and easy to set up and use. That's exactly what this is – a very useful little power supply built on an Arduino shield!

Lately, like many others, I have mostly been working from home. But unfortunately, my home workshop is not equipped to the same degree as my work office/lab. I could bring my *45V 8A Linear Bench Supply* prototype home (published in October-December 2019). It would do pretty much everything I need, but my space is limited, and it would be a rare event to make use of its full capabilities.

So I need something more compact but still useful. I decided to base it on something I already had at home, an Arduino Uno. It's capable of delivering up to 14V at a maximum of 1A. That is modest, to be sure, but handy enough for most smaller projects. And multiple units can be combined if you need several different voltages (eg, 5V and 3.3V).

Arduino considerations

Using Arduino hardware means that it would be possible to add one of many plentiful shields and modules to add a custom display or controls for the *Power Supply*. But as I already have a computer on my desk, I decided to use the existing screen and keyboard to control it.

I wrote a small computer program that controls the *Power Supply*, providing all of its useful functions without taking up valuable bench space.

Thus, the *Power Supply* can sit tucked away out of sight, with nothing more than the two output leads snaking out to wherever they are needed. The control program takes up only a small amount of screen space.

The combination of the microcontroller on the Uno and the control program allows many features to be added with no extra hardware.

For example, the control program allows five preset combinations of voltage and current to be created and instantly activated. This makes it harder to cause damage by inadvertently setting the wrong voltage or current limit.

While the *Power Supply* does not have any form of temperature sensing, it can estimate the thermal effects of a connected load to warn the user of any problems with either the load or the *Power Supply* itself.

Digital controls

Fig.1 shows the circuit of the *Mini Digital PSU*. It is effectively a 'shield' or daughterboard which plugs into the top of an Arduino Uno microcontroller board. The Uno board has an ATmega328 microcontroller, a USB-serial interface IC and some voltage regulation circuitry.

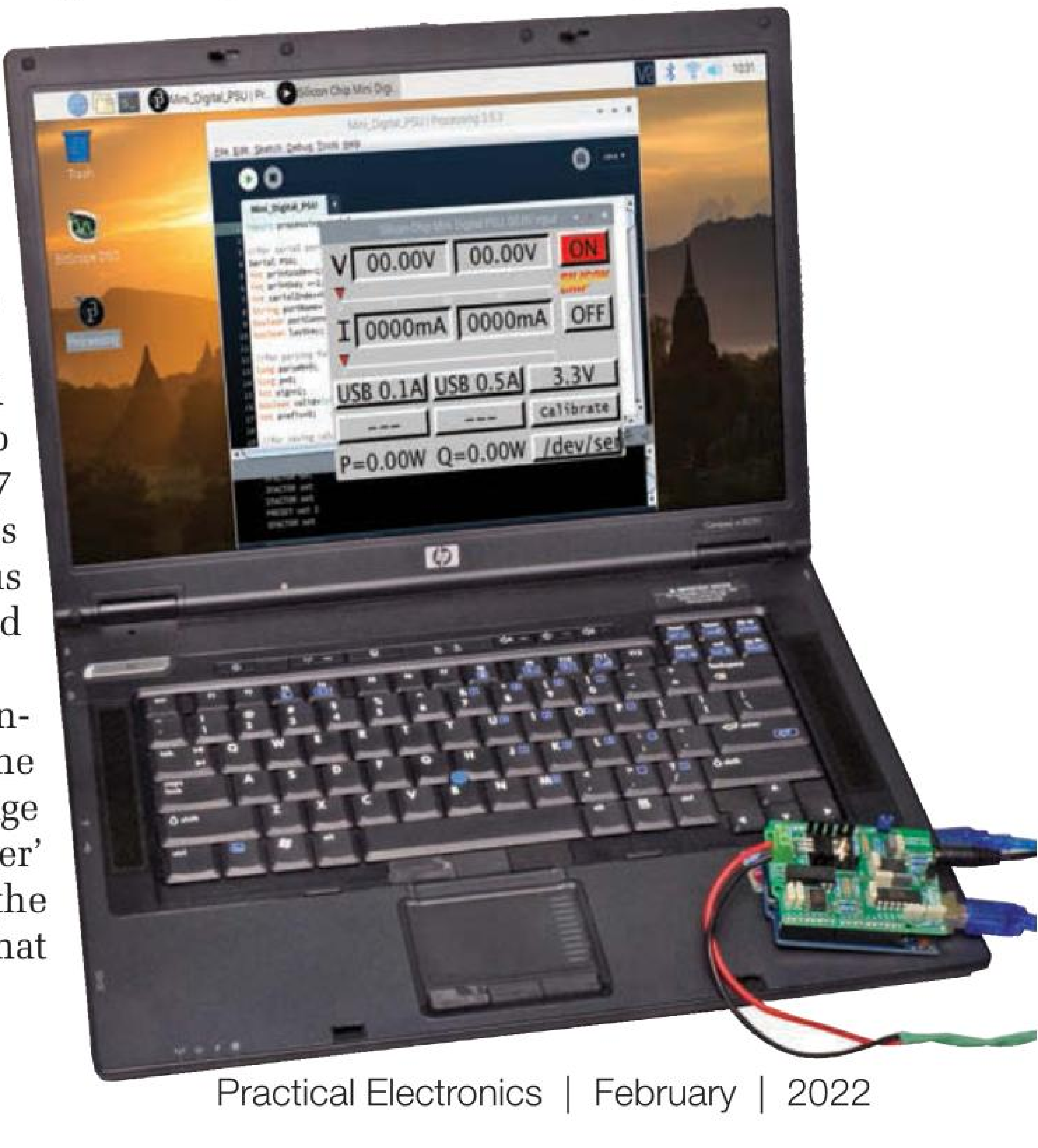
IC1 is an MCP4251 dual digital potentiometer; it contains two 5kΩ potentiometers with 257 digitally-controlled steps. This chip is controlled over an SPI bus by the Uno, from its pins 4, 13 and 11 to pins 1, 2 and 3 of IC1.

The 'tracks' of the two 'potentiometers' are grounded at one end, with a fixed reference voltage at the other end. So the 'wiper' voltages vary linearly with the programmed position, up to that reference voltage.

The voltage from pin 6 ('wiper' 1) is proportional to the desired output voltage, while the voltage from pin 9 ('wiper' 0) is proportional to the desired maximum current.

The wiper at pin 6 must be a fraction of the desired output voltage, as the digital pot IC has a maximum 5V supply voltage; hence, it can only generate voltages up to 5V.

To have a steady output voltage, we need a stable reference voltage. In this case, we're using the Uno's 3.3V rail. It comes from a practically unused 3.3V regulator on the Uno, and this is fed to IC1 via jumper JP1. This is also connected to the Uno's VREF pin, for its internal analogue-to-digital converter (ADC) peripheral to refer its readings to.



Thus the wiper of P1 (P1W, pin 6) produces a voltage in the range 0-3.3V, which is low-pass filtered by a 10kΩ/100nF RC circuit, then fed to non-inverting input pin 3 of op amp IC2. This is an LMC6482 rail-to-rail input/output CMOS dual op amp, which allows the output to go all the way down to 0V without a negative rail, and this also makes current sensing much easier (as described later).

This op amp compares the wiper voltage to a divided version of the output voltage, produced by a 51kΩ/10kΩ divider, which feeds into its pin 2 inverting input. That gives a gain of 6.1 times. Thus, around 20V at the output corresponds to the 3.3V full-scale output from digital potentiometer IC1.

The output from pin 1 of IC2 drives the base of NPN transistor Q1, which is configured as an emitter-follower. Its collector draws from the Arduino's VIN supply while its emitter feeds the supply output at CON1 via the contacts of relay RLY1 (more on this later).

This transistor effectively boosts the current capability of the op amp output so that it can supply up to 1A (from the V_{IN} supply).

The base-emitter voltage drop of Q1 is cancelled out since Q1 is in the negative feedback loop – from pin 1 of IC2, through Q1, then through the 51kΩ/10kΩ output divider back to pin 2 of IC2. Hence, IC2 adjusts its output voltage higher to achieve the set voltage at the common contact of RLY1.

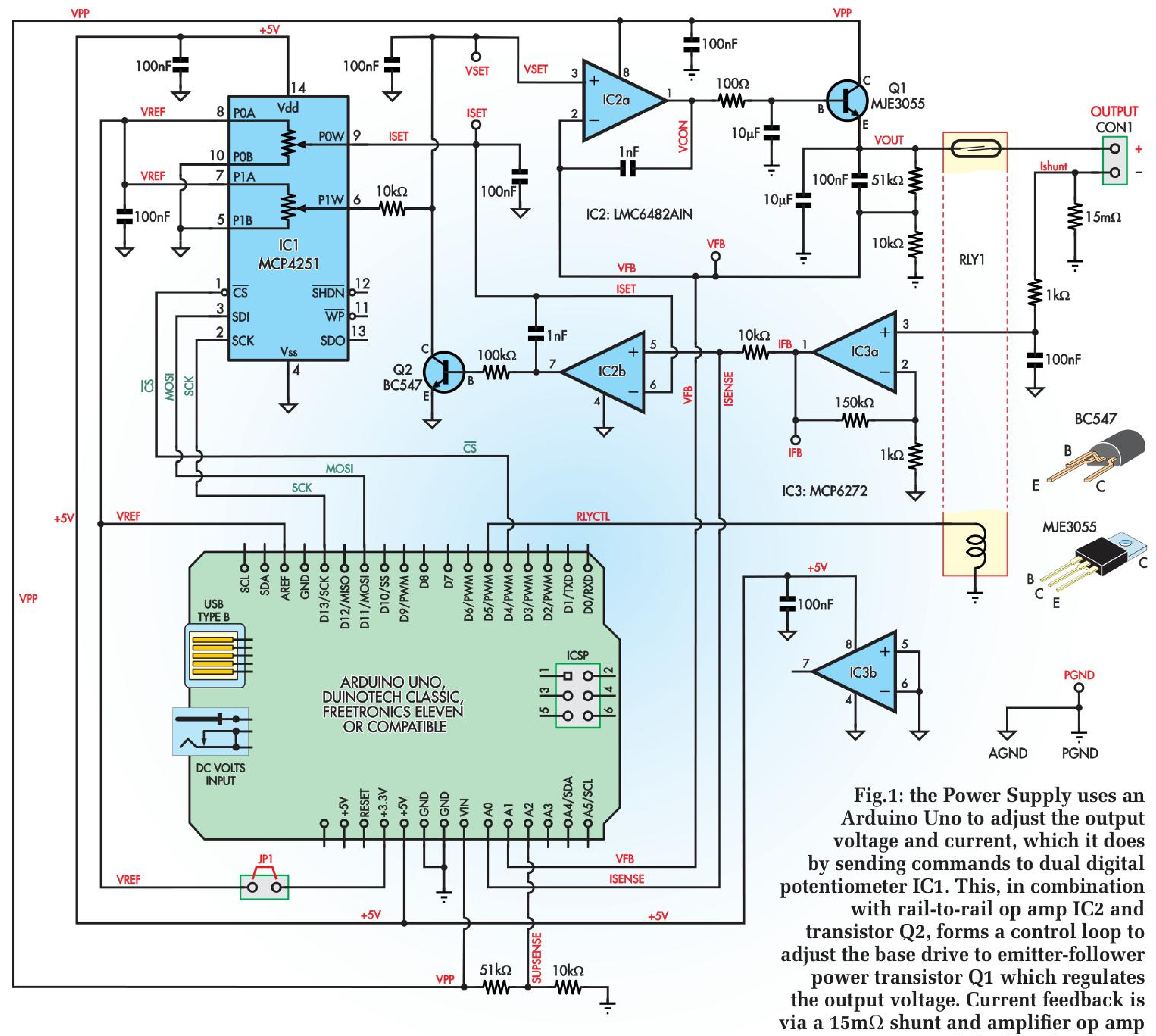
Features and specifications

- Output voltage and current: 0-14V, 0-1A
- Adjusted and monitored via a computer (eg, desktop/laptop)
- All functions under software control
- Voltage resolution: around 20mV
- Current resolution: around 20mA
- Arduino-based design means it can be expanded.

While the circuit is set up to enable an output voltage of up to 20V, in practice, other circuit elements limit the practical output voltage to around 14V. The main limit is the 5V regulator on the Arduino board, which in the case of clone boards, is only rated to 15V.

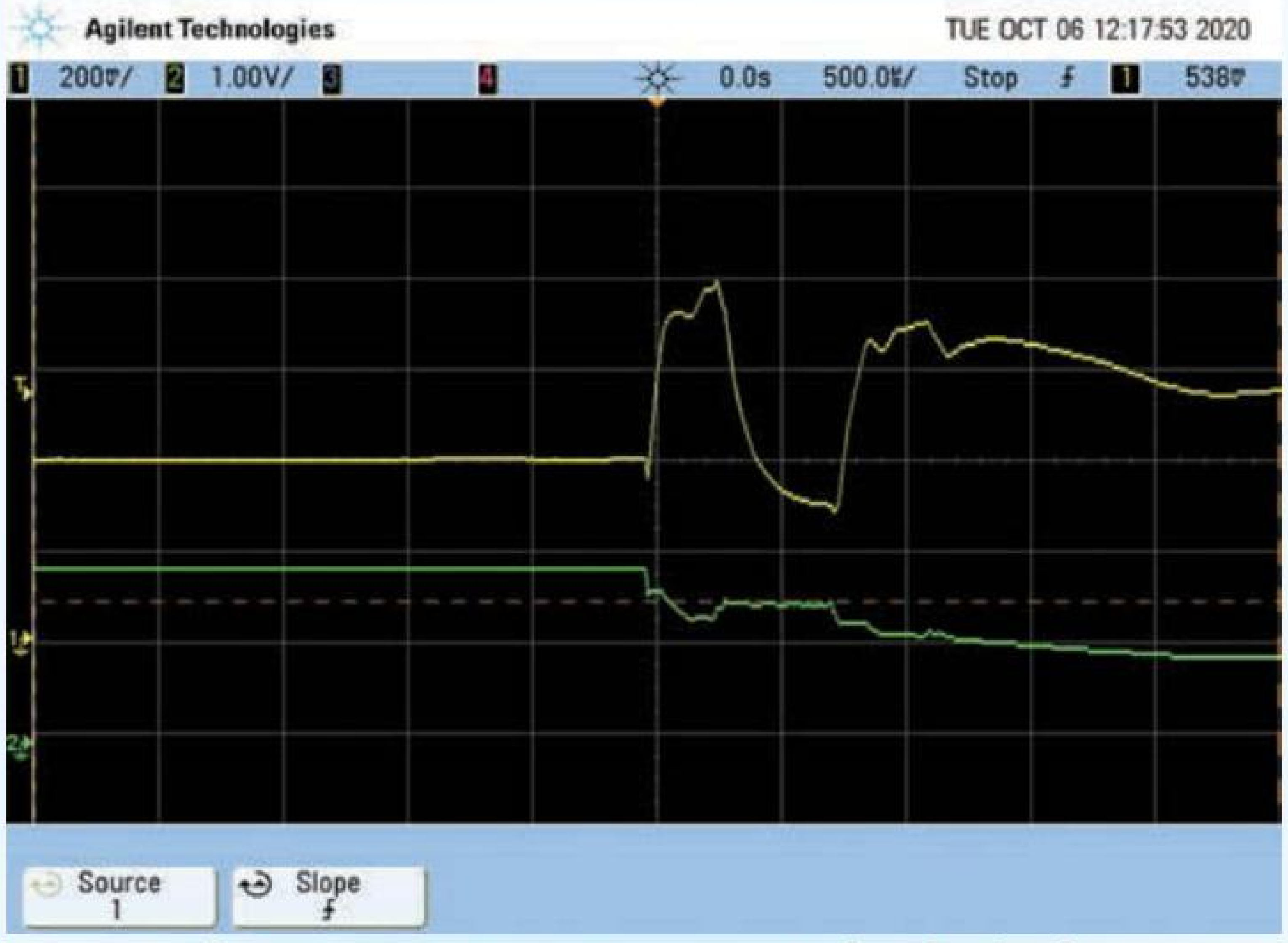
Voltage regulation

Power transistor Q1 is an MJE3055. Usually, its emitter voltage (ie, the output) is around 0.7V below its base voltage (from output pin 1 of op amp IC2). If the emitter/output voltage rises (for example, due to the load drawing less current), then

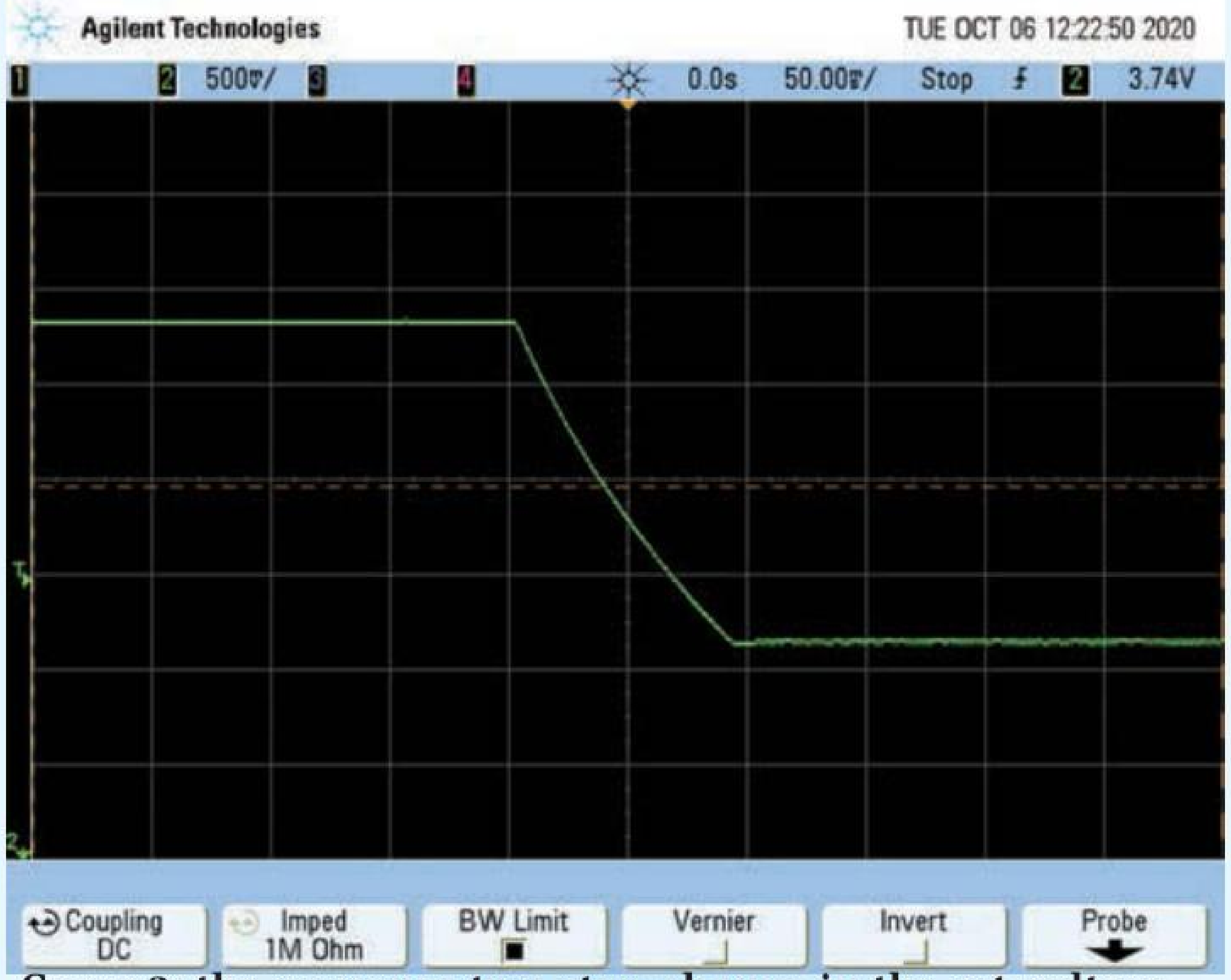

VPP
+5V
100nF
VREF
IC1
MCP4251
Vdd
P0A
P0B
P1A
P1B
P0W
P1W
CS
SDI
SCK
SHDN
WP
SDO
Vss
10kΩ
100nF
VSET
ISET
IC2a
IC2b
IC2: LMC6482AIN
1nF
VCON
100Ω
10µF
Q1
MJE3055
VOUT
51kΩ
10kΩ
VFB
Q2
BC547
100kΩ
ISENSE
IFB
IC3a
IC3b
150kΩ
1kΩ
IC3: MCP6272
OUTPUT
CON1
Ishunt
15mΩ
RLY1
BC547
MJE3055
B
C
E
MOSI
SCK
RLYCTL
USB TYPE B
DC VOLTS INPUT
ARDUINO UNO, DUINOTECH CLASSIC, FREETRONICS ELEVEN OR COMPATIBLE
ICSP
SCL
SDA
AREF
GND
D13/SCK
D12/MISO
D11/MOSI
D10/SS
D9/PWM
D8
D7
D6/PWM
D5/PWM
D4/PWM
D3/PWM
D2/PWM
D1/TXD
D0/RXD
RESET
+3.3V
VIN
A0
A1
A2
A3
A4/SDA
A5/SCL
JP1
SUPSENSE
PGND
AGND

Arduino-based Mini Power Supply

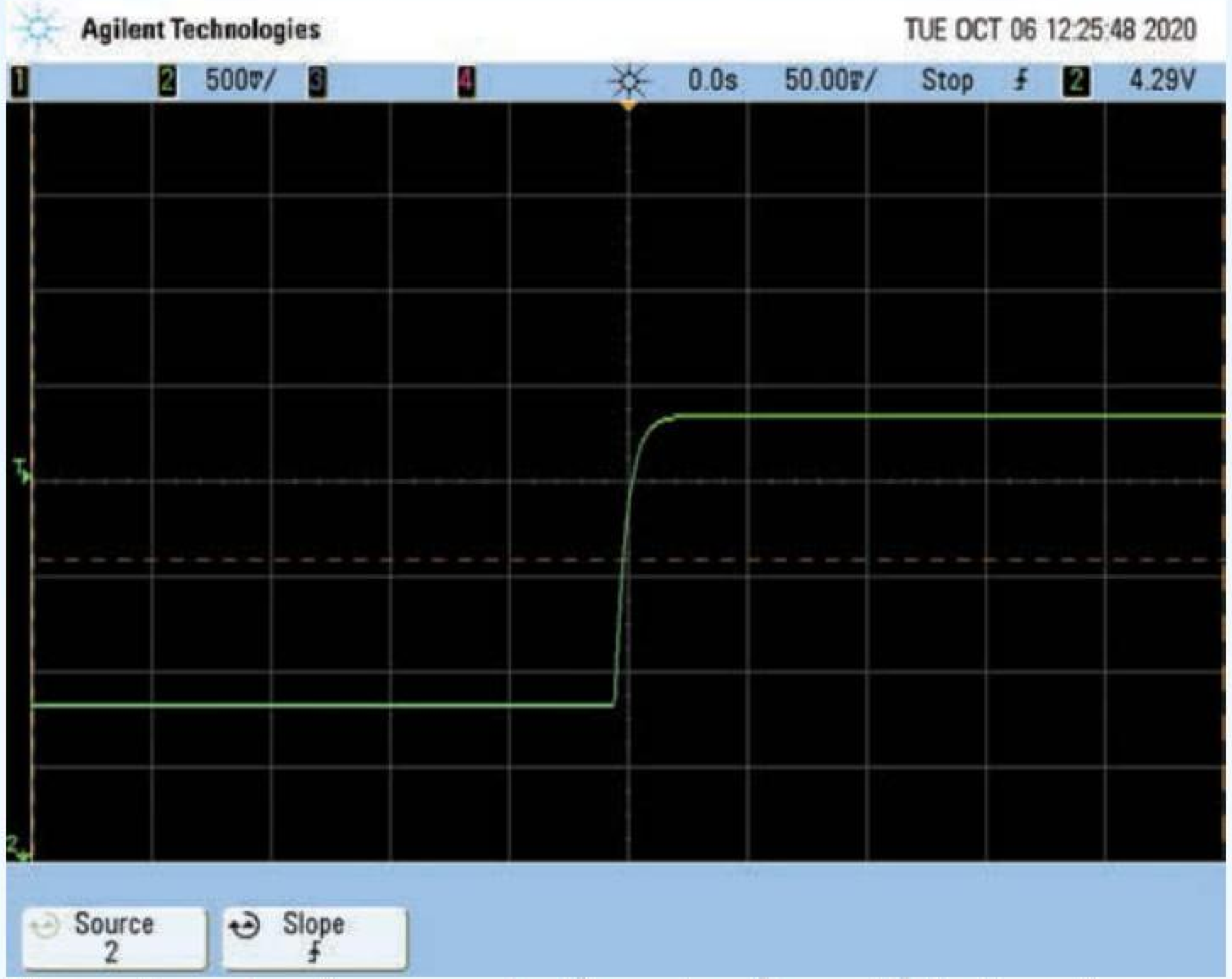
Fig.1: the Power Supply uses an Arduino Uno to adjust the output voltage and current, which it does by sending commands to dual digital potentiometer IC1. This, in combination with rail-to-rail op amp IC2 and transistor Q2, forms a control loop to adjust the base drive to emitter-follower power transistor Q1 which regulates the output voltage. Current feedback is via a 15mΩ shunt and amplifier op amp IC3, while the voltages and output current are monitored at the Arduino's A0-A2 analogue inputs.



Scope1: the response to an increase in load which triggers current limiting. The yellow trace is the voltage across the shunt resistor, so is proportional to the current, while the green trace is proportional to the output voltage. There is some current overshoot, mostly due to the output capacitance, after which the current limiting kicks in, reducing the output voltage to reach a steady-state within 1ms.



Scope2: the response to a step-change in the set voltage from 5V to 3.3V (with no load). It takes just under 100ms due to the 10µF output capacitor being discharged by the voltage-sense divider. Any significant load would speed this up dramatically.



Scope3: a step increase in the set voltage (this time from 3.3V to 5V with a 12Ω load) is much faster due to the lower impedance of the output transistor, taking just a few milliseconds.

its base-emitter voltage decreases, which starts to switch it off, causing its emitter voltage to drop.

Conversely, if the emitter/output voltage falls, the base-emitter voltage increases and Q1 turns on harder, halting the emitter voltage fall. This 'local feedback' provides a very fast response to load transients.

While the emitter-follower circuit is fairly good at tracking its input at its output, the base-emitter voltage does vary somewhat depending on the load. To overcome this, the op amp will adjust Q1's base voltage to maintain the voltage at the output voltage divider near that of the reference value on the digital potentiometer. The op amp reacts more slowly, though, due to its limited gain-bandwidth.

Transistor Q1 is fitted with a small finned heatsink, as it works as a linear pass device, dissipating any excess voltage between the supply and output. This low-profile heatsink has been chosen so another board can be stacked on top if a custom control or display needs to be added.

We have designed the shield so that it does not conflict with pins used for the *LCD Adaptor* described in June 2020, meaning we could turn this into an all-in-one unit by adding an LCD touchscreen in the future. (Although the current version of the software does not support this.)

A 10µF output filter provides modest output bypassing, which also improves transient regulation. This value is a compromise since too little output capacitance would worsen its regulation, and too much capacitance would limit the *Power Supply's* ability to quickly limit its output current under short-circuit conditions.

Between Q1 and IC2, the feedback loop has a lot of gain, so care must be taken to ensure it does not oscillate. A 100nF capacitor from the reference voltage at pins 7 and 8 of IC1 prevents transients from being seen by the op amp, which would otherwise be duplicated at the output. Similarly, the desired voltage signal at pin 3 of IC2 is stabilised with another 100nF capacitor.

There is also a 100nF feedforward capacitor across the 51kΩ upper feedback divider resistor, which reduces closed-loop gain by a factor of six or so for fast transients. Also, a 1nF capacitor is connected between the output (pin 1) and inverting input (pin 2) of IC2, limiting the op amp output slew rate. Another way of thinking about this is that it provides increased negative feedback at high frequencies. This prevents it from oscillating.

The low-pass filter formed by Q1's 100Ω base resistor and the 10µF capacitor from its base to ground also helps to stabilise the feedback loop.

Output relay

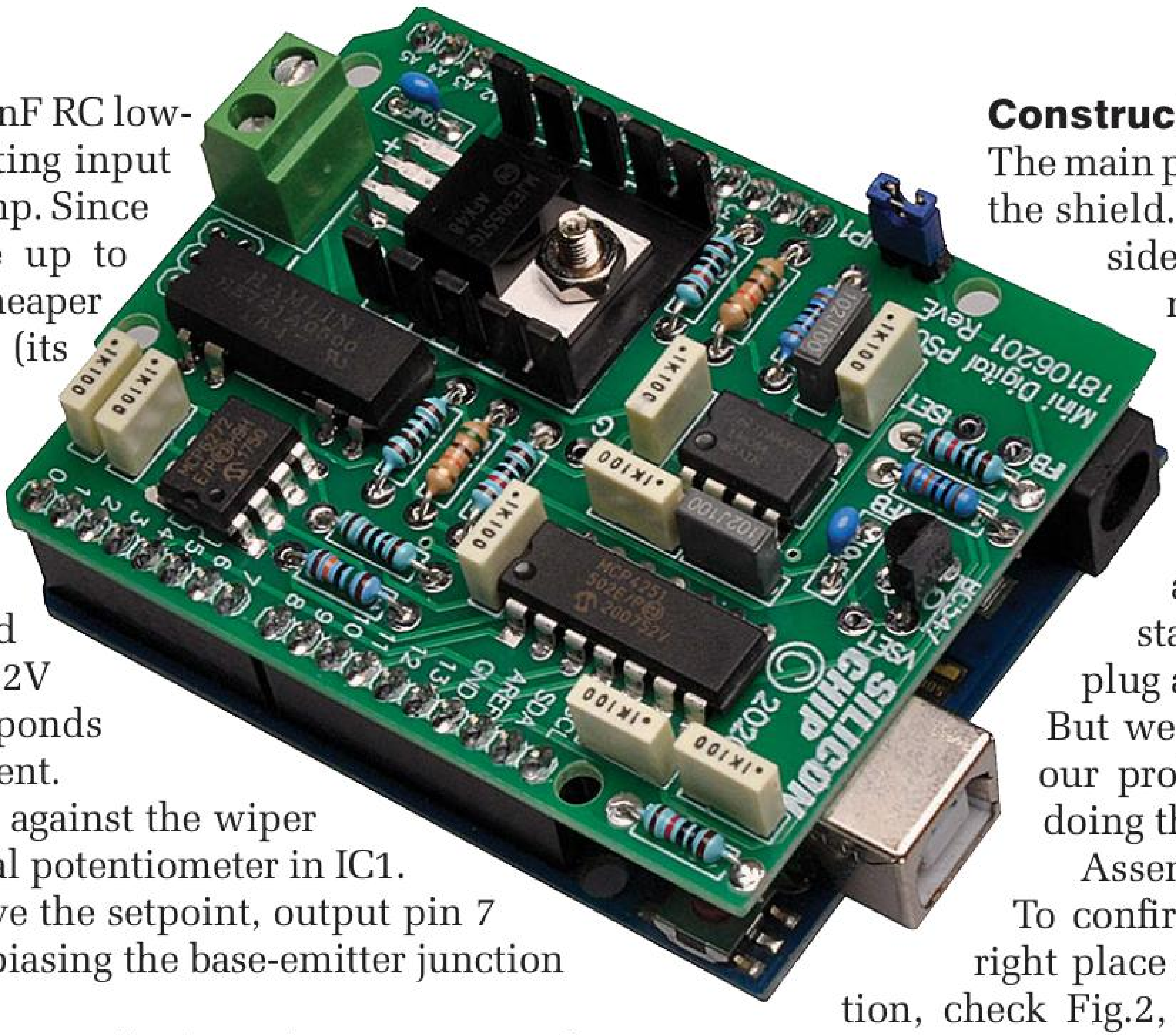
The output switching relay is a reed relay. Its coil is driven from the Arduino's D5 digital output. This is possible since the coil current of a reed relay is modest.

Unfortunately, the digital potentiometers in IC1 start with their wipers at mid-point, so a voltage will be present at the output without RLY1 disconnecting it initially. RLY1 is only energised once the regulator output voltage has settled at the desired level.

RLY1 also acts as a load disconnect switch, allowing the circuit to obtain the desired output voltage without the load being connected. It can then quickly connect the load to the already correct voltage, rather than having to ramp it up. Similarly, it can quickly disconnect the load in case of an over-current or short-circuit condition.

Current limiting

The current limiting employs a similar feedback loop to the voltage control. Here, we use the simplest current sensing possible. A 15mΩ shunt resistor in the return current path, from pin 2 of output terminal CON1 to ground, converts the load current into a voltage.



This is fed, via a 1kΩ/100nF RC low-pass filter, to the non-inverting input (pin 3) of IC3, a second op amp. Since this only needs to handle up to around 3.3V, we're using a cheaper MCP6272 dual op amp IC (its other half is not used).

IC3 amplifies the shunt voltage by a factor of 151 (150kΩ/1kΩ + 1). The amplified sense voltage is then fed to IC2's pin 5 (its second non-inverting input). So 2.2V voltage at pin 5 of IC2 corresponds roughly to a 1A output current.

This voltage is compared against the wiper voltage from the other digital potentiometer in IC1. If the output current is above the setpoint, output pin 7 of IC2b goes high, forward-biasing the base-emitter junction of NPN transistor Q2.

When Q2 is switched on, it pulls the voltage at pin 3 of IC2a down, reducing the output voltage. This should lead to a reduction in the current drawn by the load until it matches the current limit, at which point the drive to Q2 is moderated, so the output voltage should stabilise at a level where the output current is close to the set current limit.

There are a few things to note here. First, the apparent reversal of the inverting and non-inverting inputs on IC2b is because common-emitter amplifier Q2 inverts the polarity of the signal in the feedback loop. By swapping the inverting and non-inverting inputs, we effectively re-invert it and get the correct polarity.

Also, like the voltage feedback loop, stability is improved by a 1nF capacitor between the output (pin 7) and inverting input (pin 6), plus there is a 100nF capacitor stabilising the current set voltage at pin 6.

The voltage and current feedback signals also go to two of the analogue-capable pins on the Uno board. Thus the Uno can sense (with its ADC peripheral) the voltage and current using pins A1 and A0 respectively.

The V_{IN} supply voltage is measured via a second 51kΩ/10kΩ divider at analogue input A2. That allows the micro to calculate the voltage drop across Q1, and infer its thermal dissipation.

On the PCB, there are test points for the four sense/reference voltages. These are labelled VFB, IFB (voltage and current feedback), VSET and ISET (voltage and current setpoints), plus one for GND.

Arduino software

The Arduino firmware produces SPI data to set the desired voltage and current limits, then closes the relay to enable the output when prompted by the user. The hardware on the shield then manages the output voltage, reducing it if the current limit is reached as described above.

Once the voltage and current are set, the regulator operation is automatic; it does not depend on the software for control.

The microcontroller measures the supply and output voltages, and load current, then sends this data to the program running on your computer for display.

Calibration consists of determining the exact relationship between digital values (ADC readings and digital potentiometer settings) and the resulting analogue voltages. These coefficients can be calculated from measured component values.

The *Power Supply* will be fairly accurate 'out of the box'. But its accuracy can be improved by taking readings with a multimeter, determining the exact ratios and programming these into the code. A calibration routine in the PC program simplifies this process, automatically calculating the new ratios from measurements.

Construction

The main part of the assembly is building the shield. The parts all fit on a double-sided PCB coded 18106201, which measures 69mm x 54mm and is available from the *PE PCB Service* – see Fig.2.

The first decision to make is whether you want to build it with plain headers or stackable headers. You will need stackable headers if you plan to plug any shields on top of this one. But we used regular pin headers on our prototype, as we don't plan on doing that immediately.

Assembly is then straightforward. To confirm everything is going in the right place and with the correct orientation, check Fig.2, the PCB silkscreen and the matching photos as you fit the parts.

Start by fitting the 15mΩ surface-mounted resistor, which goes on the underside of the PCB. Some constructors like to use a wooden clothes peg to hold an SMD component in place while soldering it.

Flip the board over and tack one lead in place with your iron. If the part is flat and square within the silkscreen markings, solder the other lead. Otherwise, remelt the first pad and adjust the resistor, using tweezers if necessary, until it is placed correctly. Then solder the second lead and flip the PCB back over.

Fit the 11 through-hole resistors on the top of the PCB, as indicated by the silkscreen markings. Check their values with a multimeter, as some of the markings can look quite similar.

Follow with the eight 100nF and two 1nF capacitors, which should be marked with their values (or codes representing them, like 104 and 102 respectively). None of those are polarised; nor are the 10µF capacitors which can be through-hole or SMD types. Mount them now.

Next, install the smaller transistor, Q2. Crank the leads to fit the PCB pads, ensuring that when mounted, the body sits low in case you need to add a shield above this one. Ensure that it matches the outline on the PCB silkscreen.

Follow with the TO-220 transistor, Q1. It is mounted on a finned heatsink. First, bend the leads backwards by 90° around 7mm from the transistor body, then thread the leads through the PCB pads. Check that the larger mounting hole is aligned and adjust the leads if necessary.

Remove Q1 from the PCB and insert the M3 machine screw through the back of the PCB. Add the heatsink on top, then the transistor and thread on the nut. Before tightening, ensure that the heatsink and transistor are square within the footprint. Carefully tighten the nut (to avoid damaging the transistor leads), then solder its leads and trim them.

Most of the remaining parts are in DIL packages. Avoid using IC sockets, as not only will they have a worse connection than direct soldering; they will also cause the components to sit much higher.

RLY1 has eight pins but comes in a 14-pin size package. It sits above Q2; the notch in its case faces to the right. Gently bend the leads to line up with the pads and fit them. Solder two diagonally-opposite leads and check that the part is flat; adjust if it is not. Solder the remaining leads and then go back and refresh the solder on the first two leads.

IC1 is a 14-pin part; its pin 1 notch should butt right up to the adjacent capacitor. IC2 is an LMC6482, as marked on the silkscreen. Do not mix it up with IC3, which is specified as an MCP6272, although you could use another LMC6482 instead.

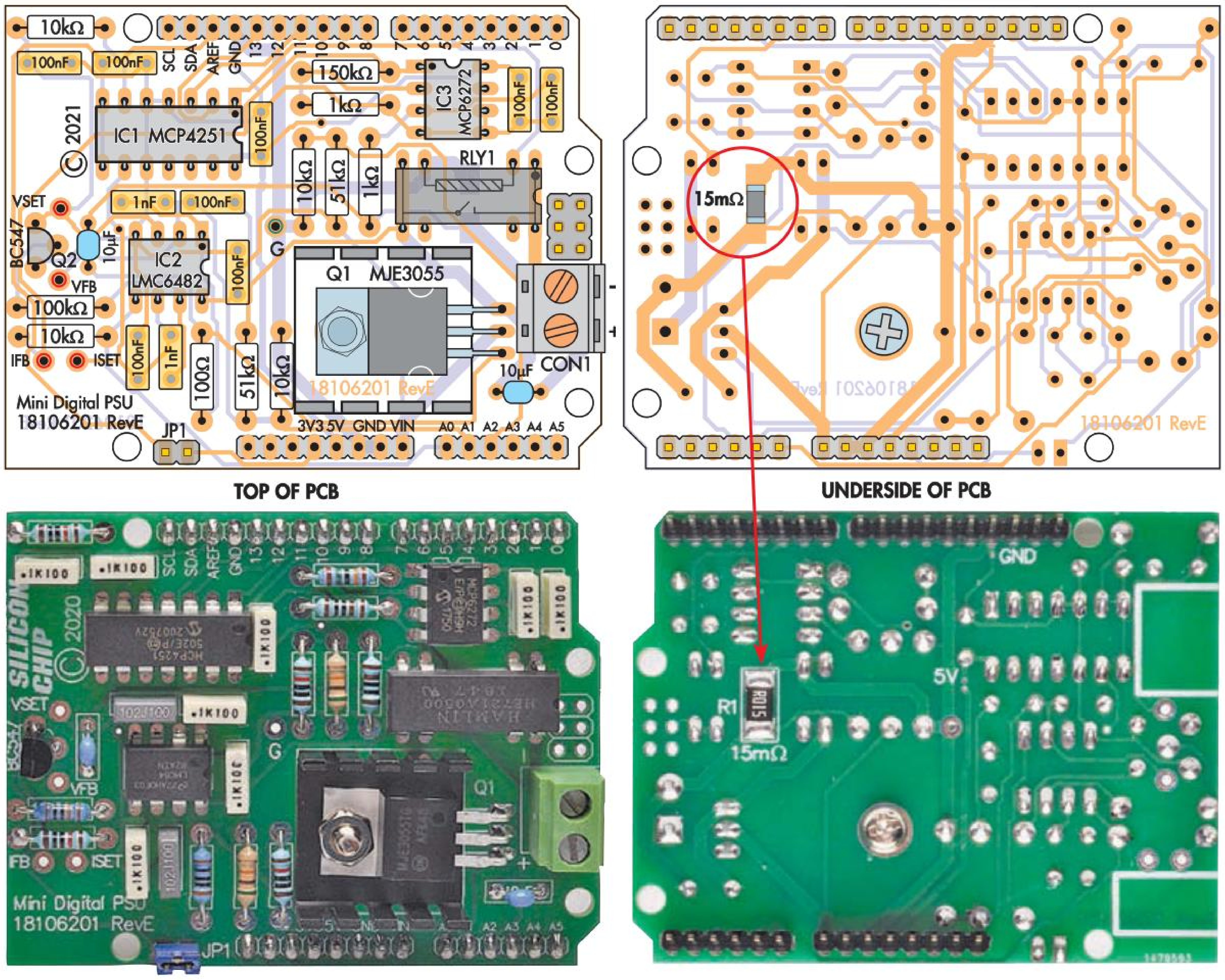


Fig.2: this deceptively simple Arduino shield turns an Uno into a regulated bench Power Supply. Apart from the pin headers, the only component on the underside (and the only SMD) is the 15mΩ shunt resistor. Power transistor Q1 has a small heatsink as it can dissipate several watts. The ICs, relay and transistors are polarised so must be oriented as shown, while the other components can go in either way around. Several test points are provided, but they are not needed for calibration.

And to further assist in construction, here are the matching same-size photos of the shield, from both sides.

Use a similar technique as RLY1 to fit IC1, IC2 and IC3. Once that is done, check for any bridges or dry solder joints and repair as necessary by using a solder sucker or solder braid to remove excess solder. Apply the iron and fresh solder to finish the solder joint.

Headers and jumper

Attach the Arduino mounting headers, along the edges of the board, next. If you are using male headers, then fitting them is straightforward. Use the Uno as a jig and plug the pin headers into the Uno, then place the PCB on top. After checking that everything is flush and square, solder the pin headers from above and unplug the assembly from the Uno.

If you want to use stackable headers, then it is a bit trickier, although the Uno can still be used as a jig. In this case, the headers thread through the PCB from above and into the Uno. Flip the assembly over so that the *Power Supply* PCB is at the bottom.

Now you have access to the pins of the stackable headers from below. That should be sufficient to tack the endmost pin of each strip to keep the headers in place. Check that the headers are flat against the PCB and adjust if needed.

Unplug it from the Uno to give better access to the remaining pins. Solder them, then refresh the end pins.

In this case, you will probably also need to solder a two-by-three pin stackable header block to the R3 header location on the board, to pass those signals through to a board stacked above.

JP1 consists of a male header and jumper shunt. Fit the shunt to the header, slot it into the PCB and solder its pins. The shunt will keep the pins in place even if the plastic shroud melts a little.

Finally, mount the output connector, CON1. We used a two-way screw header, although you might prefer something different depending on how you want to use the *Power Supply*.

Solder CON1 in place and then fit the PCB to the Uno. Unless the Uno is new and unprogrammed, you should remove JP1, in case the existing sketch uses a different voltage reference which could conflict with the 3.3V supply and possibly damage it.

Software

There are two elements to the software of this project – the first is the firmware that runs on the Uno. The second is the computer application that interfaces with it. The Arduino firmware 'sketch' is available for download from the February 2022 page of the *PE* website.

We're assuming that you have some familiarity with the Arduino IDE (integrated development environment), although it isn't too hard to figure out if you're new to it. The IDE can be downloaded for free from: **https://www.arduino.cc/en/software**

We're using version 1.8.5, but practically any version should be fine as the sketch is quite simple and doesn't need any special libraries.

With that installed, the next step is to load the Uno with the firmware. Connect the Uno to a USB port, select the Uno's serial port from the Arduino IDE Tools menu, then ensure that the Uno board is selected as the target (Tools -> Board -> Arduino Uno). Press Upload, and once the sketch has uploaded, insert JP1 and open the Serial Monitor at 115,200 baud (CTRL + SHIFT + M in Windows).

The sketch is fairly simple; it listens on the serial port for commands like 'V100', 'I50' or 'R1' to set the voltage, current or the relay state respectively. Since the communication to and from the *Power Supply* is simply over a serial line, we can also test the unit by typing commands into a serial terminal program such as the Serial Monitor.

Such a simple scheme means that it can be manually controlled if necessary. But it also means the *Power Supply* can be very easily controlled by other software; it just has to send the correct commands and process the (simple) responses.

Even if no 12V supply is available, the Uno itself will feed around 4V to the VIN pin (and thus the *Power Supply*) for testing. This is enough for us to do some simple, low-voltage testing to check that the unit works as expected.

Testing

With the *Power Supply* plugged in via USB and the Serial Monitor open, you should see a stream of lines showing values prefixed by J, U and S. The J and U values should be close to zero, but S will be around 200 (indicating around 4V at VIN). To test the relay, type 'R1' or 'R0' followed by Enter. You should be able to hear it gently clicking on (after R1) and off (after R0).

You can send commands to the digital potentiometer by typing either V or I, followed by a number in the range of 0-256, then enter.

These numbers are the raw digital potentiometer values, as all calibration is done on the host computer program. With the relay on and both the V and I values set to non-zero values, you should measure a voltage across the output terminals.

The J, U and S values are raw ADC readings (0-1023) of the input and output voltage and current, taken several times per second by the Uno. The J, U and S letters chosen are to avoid confusion with the commands V and I. The host program converts the 0-1023 readings to real-world voltages and currents.

To test the output with a multimeter connected to CON1, enter the command 'R1', followed by 'V255' and 'I255'. This should allow the output to get within about 0.7V of the VIN supply voltage (limited by the inherent diode drop of the emitter follower Q1).

Try some lower values for V (eg, V25) to check that the output can be regulated to a lower level. That should give you about 2V, while V37 should give about 3V and V13 should give about 1V. To check higher output voltages, you will need to connect a 12-15V supply to the Arduino's barrel socket (but watch that upper voltage limit!).

For this testing, it would be a good idea to connect the Uno to your computer via a *USB Port Protector*, like our design from May 2019. That will mean that even if there is a fault in your *Power Supply* that results in 12V or more being fed back to the USB signal pins (which operate at 3.3V), it shouldn't damage your computer.

Processing app

We wrote the computer control app in the Processing language. The Processing IDE is available on Windows, Mac and Linux (including the Raspberry Pi). Using the IDE, you can run the program or compile it to a standalone executable file for your system. It's based on Java, so you will probably need a Java runtime environment (JRE) installed to run the program.

Processing can be downloaded for free from here: **https://processing.org/download/** (we used version 3.5.3).

There are no special libraries or add-ons needed. Open the Processing sketch (a file with a .pde extension) using the File menu and run it using the Ctrl-R key combination. A standalone executable can be created from File -> Export Application.

Referring to Screen1, the actual and set voltages and currents are shown as bar graphs and in digital form at the top of the window. A similar display below shows the actual and set currents. Two large buttons are provided to turn the output on and off. Below this are five preset buttons and a button to access the calibrations page.

Along the bottom are displays for output power (P) and transistor Q1 power (Q). These change colour as the power increases. At bottom right is an indicator for the serial port.

The initial calibration of this software comes from our prototype, so it should be roughly correct within component tolerances. It's easy to fine-tune it, though.

Using it

Press '+' and '–' on your keyboard to cycle through the available serial ports. When the Uno's port is selected, press 's' to connect – if the connection is successful, the serial port will turn green. If it does not connect, check that the port is not in use by another program (eg, the Arduino Serial Monitor).

Parts list – Arduino-based Power Supply

1 double-sided PCB coded 18106201, 69mm x 54mm, available from the PE PCB Service
1 Arduino Uno or compatible board
1 12V-15V 1A plugpack with 2.1mm DC plug to suit the Uno, or a similar power source
1 2-way screw terminal (CON1)
1 6-way pin header (or stackable header, see text)
2 8-way pin headers (or stackable headers, see text)
1 10-way pin header (or stackable header, see text)
1 TO-220 finned heatsink (for Q1) [Jaycar HH8502]
1 2-way pin header and jumper/shorting block (JP1)
1 2x3-way stackable header (optional; needed if another shield to be attached above)
1 5V coil DIL reed relay (RLY1) [eg, Altronics S4100, Jaycar SY4030] *supplies built with the Jaycar relay should set the current limit no higher than 500mA to avoid damage to the relay, due to this relay only having a 500mA switch rating*

Semiconductors

1 MCP4251-5k 5kΩ dual digital potentiometer, DIP-16 (IC1) [Digikey, Mouser]
1 LMC6482 dual op amp, DIP-8 (IC2) [Jaycar ZL3482]
1 MCP6272 dual op amp, DIP-8 (IC3; LMC6482 can substitute)
1 MJE3055 10A NPN transistor, TO-220 (Q1) [Jaycar ZT2280]
1 BC547 100mA NPN transistor, TO-92 (Q2) [Jaycar ZT2152]

Capacitors

2 10µF 16V leaded X7R ceramic (or SMD M3216/1206-size)

8 100nF MKT	(code 103, 100n or 0.1)
2 1nF MKT	(code 101, 1n or .001)

Resistors (all 1/4W 1% axial metal film except where noted)

1 150kΩ	(brown green black orange brown or brown green yellow brown)
1 100kΩ	(brown black black orange brown or brown black yellow brown)
2 51kΩ	(green brown black red brown or green brown orange brown)
4 10kΩ	(brown black black red brown or brown black orange brown)
2 1kΩ	(brown black black brown brown or brown black red brown)
1 100Ω	(brown black black black brown or brown black brown brown)

1 15mΩ 1% SMD, M6532/2512-size [SC ONLINE SHOP SC3943]

The 's' key has a toggle action, so it can also be used to disconnect from the *Power Supply*.

Drag the arrows on the bar graphs with the mouse pointer to set the voltage and current. The green arrow is the setpoint, which corresponds to the leftmost digital display. The red arrow and rightmost numbers correspond to the actual voltage and current values.

Click the 'ON' button to energise the relay and enable the output. Note that the PSU reads the voltage before the relay, so it will show a value even if the relay is off. The 'ON' button turns green when the relay is on. Use the 'OFF' button to shut it off.

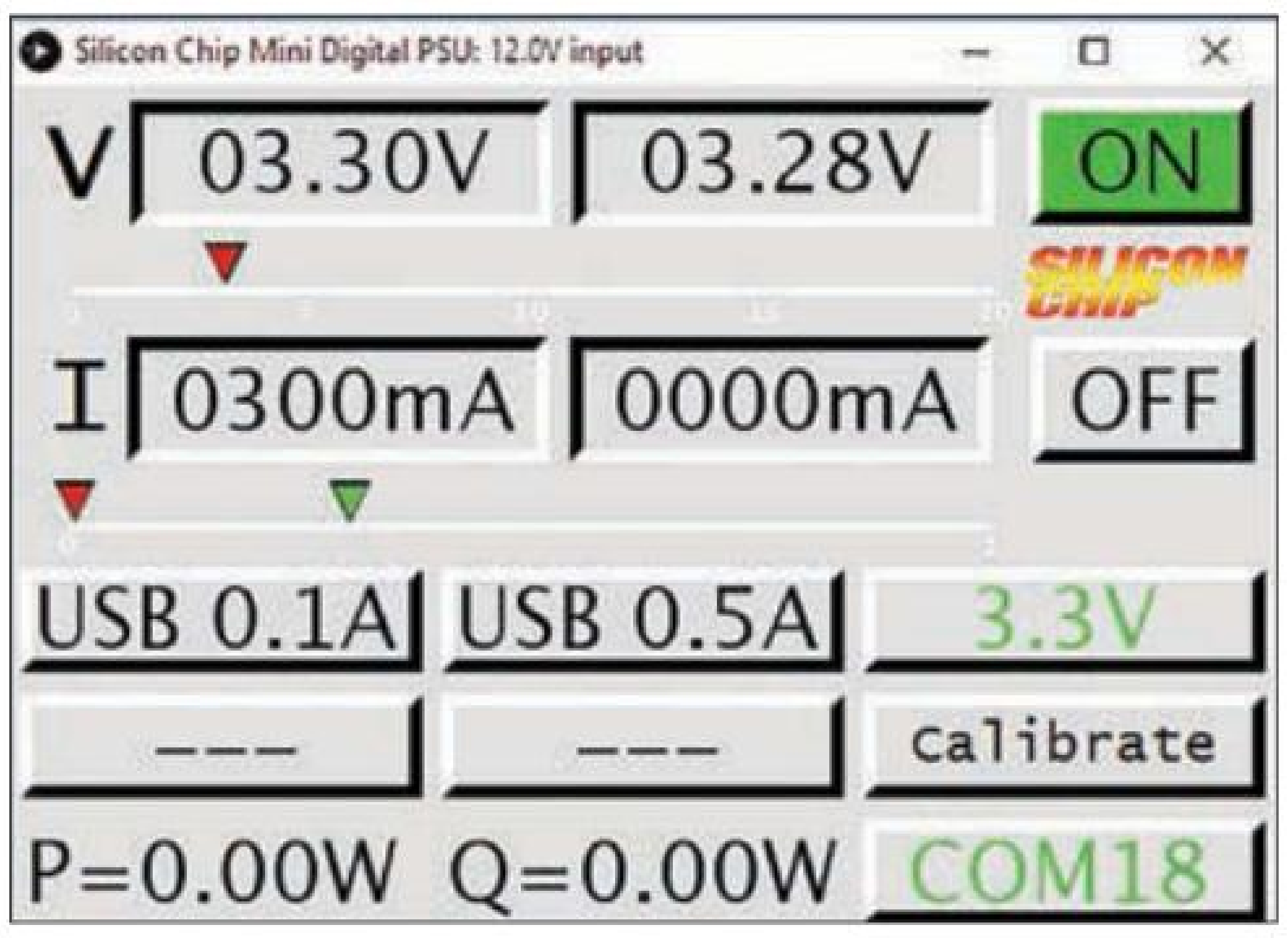
Pressing any of the five preset buttons will load that preset into the voltage and current setpoints. In Screen1, preset three is loaded, so its button is highlighted.

Calibration

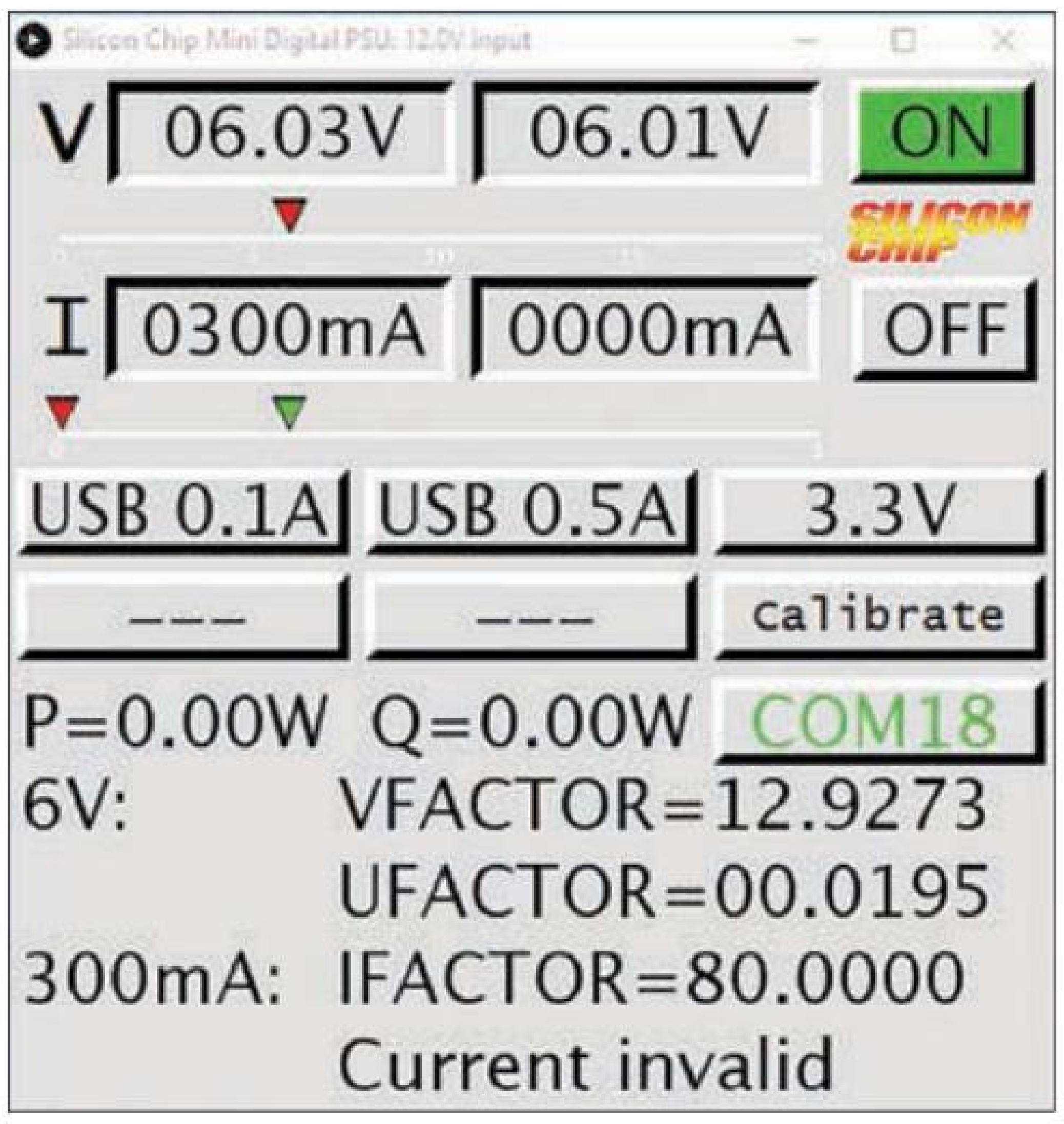
Pressing the 'Calibration' button will expand the window to show the calibration values (see Screen2). Our copy of Processing stalls for a few seconds when this happens; it is a known bug which will hopefully be fixed in a later version. To close the Calibration view, click in the lower part of the window.

Calibration is achieved in two stages. The first is to calibrate the voltage, which requires a voltmeter to be connected across the *Power Supply* output (CON1).

Turn on the output and set the current to any value above zero; this ensures the current limiting doesn't kick in, which would reduce the output voltage.

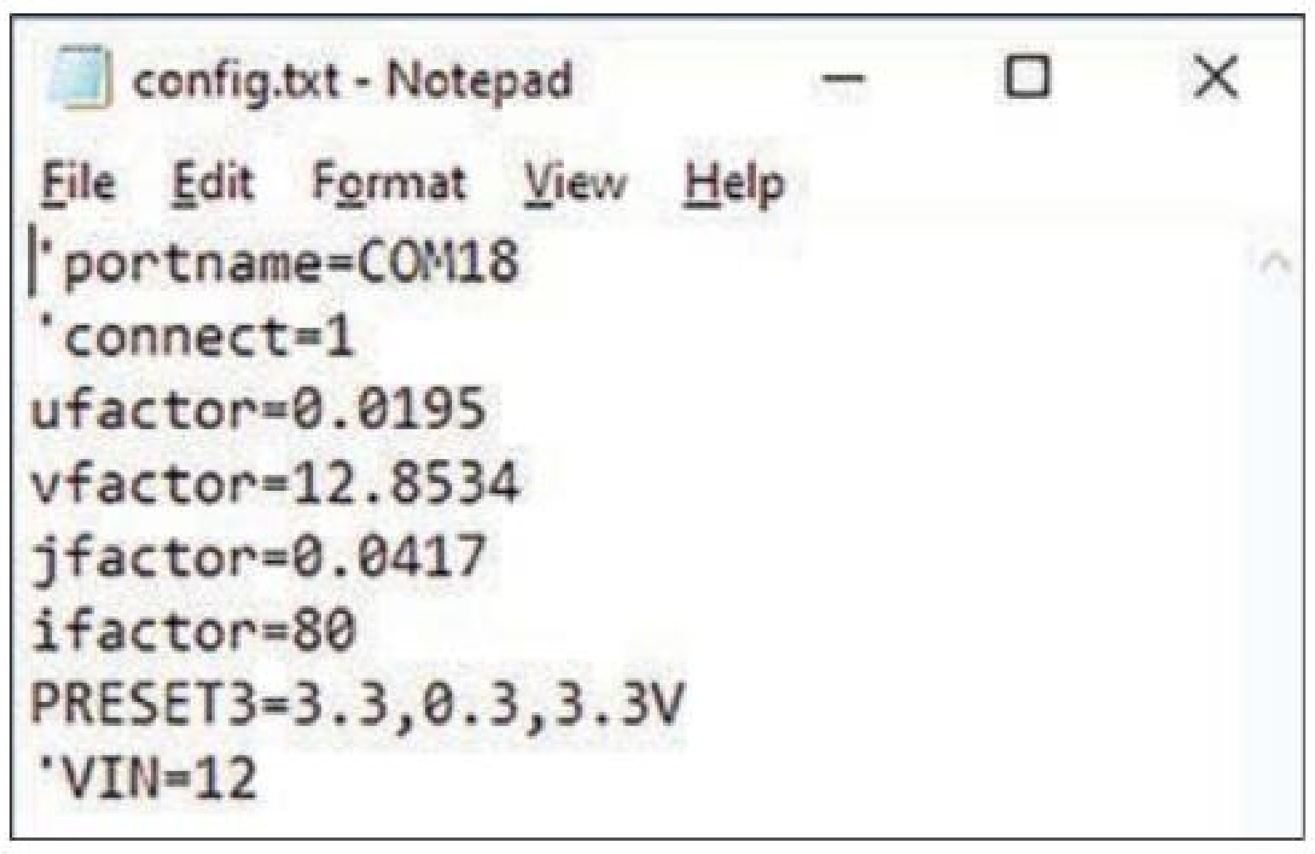


Screen1: our Processing application provides slider controls for voltage and current at the top, along with simple switches to switch the output on and off. Presets are displayed and selected below, along with power information. The incoming supply voltage can be monitored in the title bar.



```
P=0.00W  Q=0.00W
6V:      VFACTOR=12.9273
         UFACTOR=00.0195
300mA:   IFACTOR=80.0000
         Current invalid
```

Screen2: the calibration procedure is simple. You adjust the controls until the multimeter reading matches the voltage and current readings shown at lower left, after which you simply copy the parameters to the configuration file.



```
'portname=COM18
'connect=1
ufactor=0.0195
vfactor=12.8534
jfactor=0.0417
ifactor=80
PRESET3=3.3,0.3,3.3V
'VIN=12
```

Screen3: the 'config.txt' file contains calibration parameters and up to five named presets. You can also set the serial port and whether the application should automatically connect to it at startup.

Next, adjust the voltage slider until the multimeter reads as close to 6V as possible. A 12V-15V DC external supply is ideal for doing this, but even 9V DC would be sufficient. Note that the two pointers may not line up to 6V. This is expected, as we are still calibrating the unit.

Now, write down the 'VFACTOR' and 'UFACTOR' values that are displayed in the bottom panel.

To calibrate the current side, turn the output off and switch your multimeter into a mode and range capable of reading up to 400mA. You will probably need to change how the meters leads are plugged in too.

Since your multimeter is effectively forming a short circuit, you can include a power resistor in series with the multimeter leads for extra protection, and to reduce dissipation in the output transistor. For example, a 10Ω 5W resistor would work well.

Switch on the output and move the current pointer up until the multimeter reads 300mA, then note down the lower ('IFACTOR' and 'JFACTOR') calibration values and turn the output off. Be quick about this, as the transistor can get quite hot during this stage.

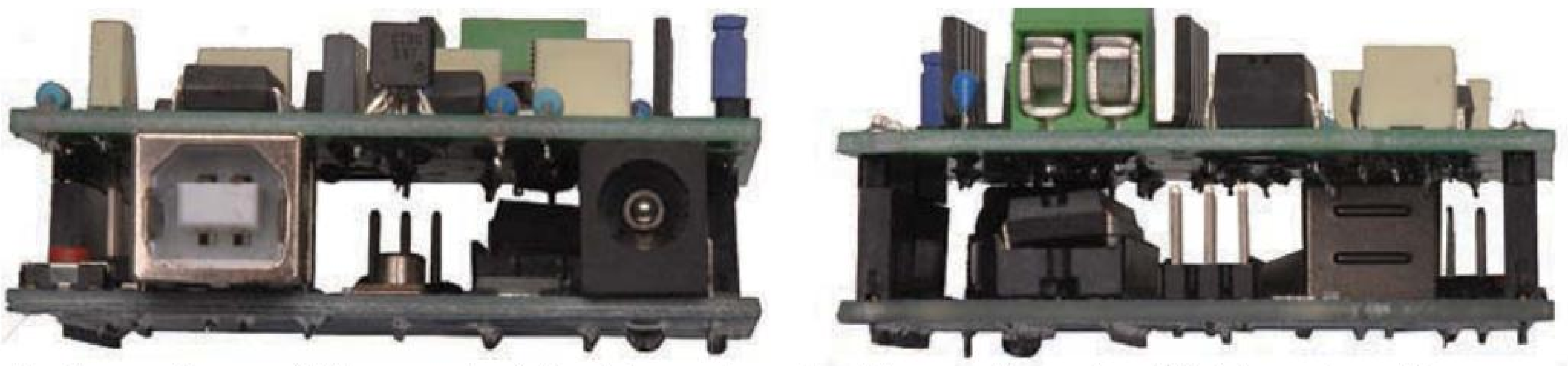
Configuration

The calibration factors (along with other settings) are stored in a file called **config.txt**. This must be in the same folder/directory as the .pde file for the Processing sketch. Open it and add or modify the four calibration factors you wrote down. The result should look like that shown in Screen3. Note that the app does not care about upper or lower case in these settings.

You'll need to restart the program to load the new configuration. If you are running it from the Processing IDE (rather than an exported app), you should see that the calibrations are loaded in the log window at the bottom, like this:

```
UFACTOR set
VFACTOR set
JFACTOR set
IFACTOR set
```

If these are not seen, then there may be an error, and the values have not been loaded.



End-on views of the sandwiched boards – the Power Supply shield on top; the standard Arduino Uno (or compatible) below.

The configuration file also supports some other options. SFACTOR is used for calculating VIN; it is theoretically (within component tolerance) the same divider as that for UFACTOR, so you can use the UFACTOR value here too. It's only used for display and dissipation calculations, so isn't as critical as the other values.

It is a simple scaling factor from the raw ADC result (0-1023) to voltage, so can also be adjusted by comparing with a multimeter reading. For example, if the displayed supply voltage is 1% too low, then increase SFACTOR by 1%.

You can also set the default serial port and whether or not it should connect when you run the program with the PORTNAME and CONNECT parameters. The nominal supply voltage can also be provided with the VIN parameter.

The PORTNAME should be set before the CONNECT line so that the correct port is opened. The naming scheme for ports will differ between operating systems.

The five presets are set with PRESET1 to PRESET5, with the values being voltage (in volts), current (in amps) and name (cropped past seven characters). These parameters are separated by commas.

Naturally, all configuration variables have reasonable defaults in case the configuration file is missing or empty. We've left a few potential lines in the file prefixed by an apostrophe; the program ignores these lines until you remove the apostrophes.

Usage

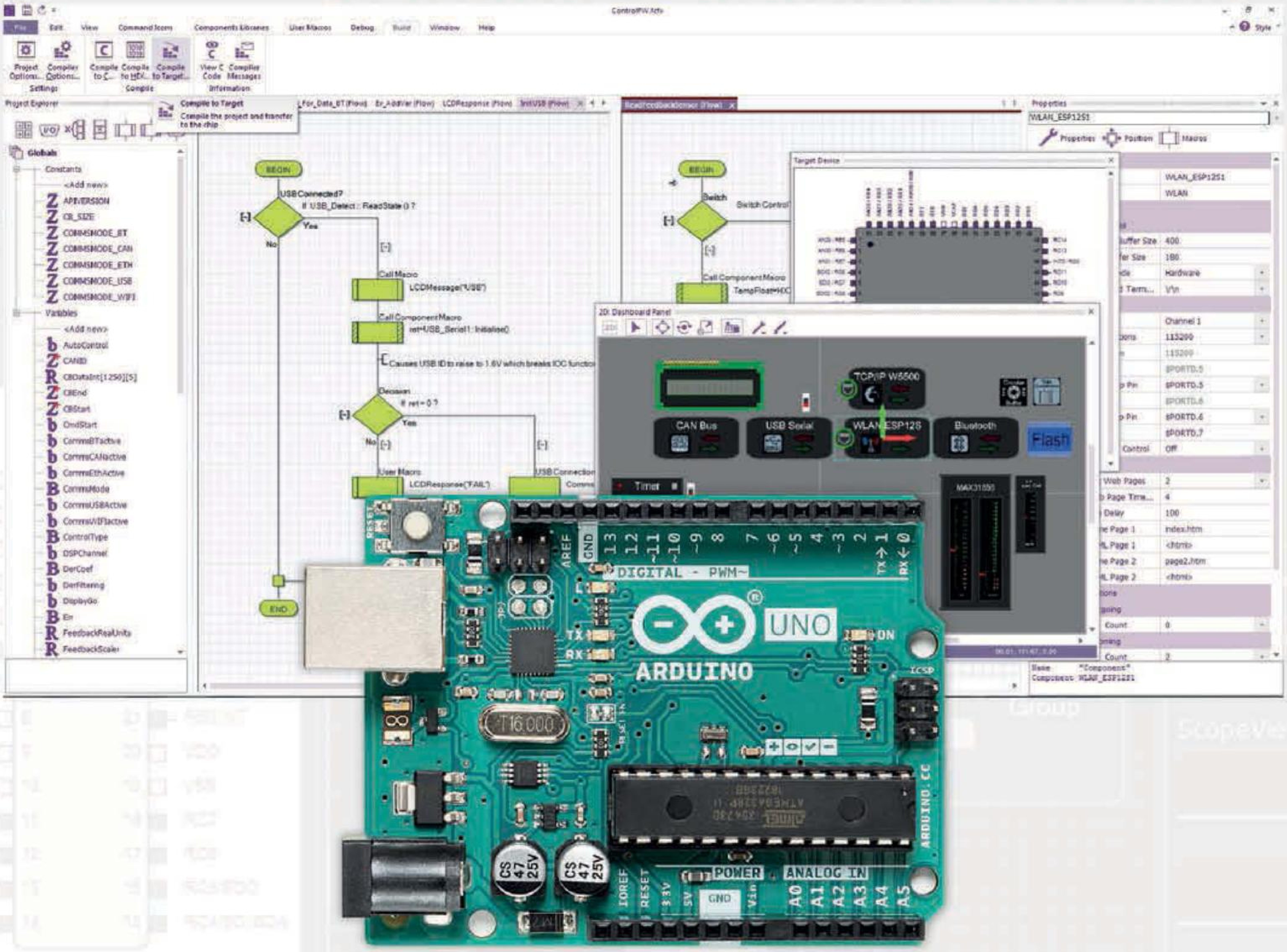
The *Power Supply* control app has been designed so that using it should be intuitive. We reckon that this way, it is much easier to use than a supply with physical controls like a few pots and a small display.

It is by no means a high-accuracy piece of test gear but still very handy to have on your desk, especially since it doesn't take up much space.

We haven't described how to fit it into any sort of enclosure, as you really can just use it as-is.

If you do want to enclose it, a UB3 Jiffy Box is the simplest and cheapest option, and its generous size should allow some airflow for cooling. A pair of holes in each end will be sufficient to run all the necessary leads.

Reproduced by arrangement with
SILICON CHIP magazine 2022.
www.siliconchip.com.au



CREATE FANTASTIC ELECTRONIC SYSTEMS
USING

FLOWCODE

20% off your Flowcode purchase using code: EPE20
Use code at checkout: flowcode.co.uk/buy

Off grid? On grid with battery backup? How do you monitor the state of your batteries?

Battery Monitor Logger

By TIM BLYTHMAN

Knowing the condition of your batteries is essential for keeping them healthy longterm. A system that can monitor and log vital battery statistics is a great aid, and can help you to avoid having to shell out for expensive replacements. It can also be used for troubleshooting, such as when you don't know which device is responsible for periodically discharging a battery.

Solar and wind power is growing in use and getting cheaper, so there is a need to maintain batteries associated with such systems. You might also have a large battery in a shed, caravan, boat or another vehicle that you need to monitor. Backup batteries for mains power failures are another case where you might need a battery monitor or logger.

Our new *Battery Monitor Logger* is versatile and capable, being able to handle a charger and two separate loads out-of-the-box. It is based on a *Micromite LCD BackPack*, so can be reprogrammed in MMBasic, Micromite's variant of the BASIC language. But as we have written software with many useful features, you don't need to do any programming.

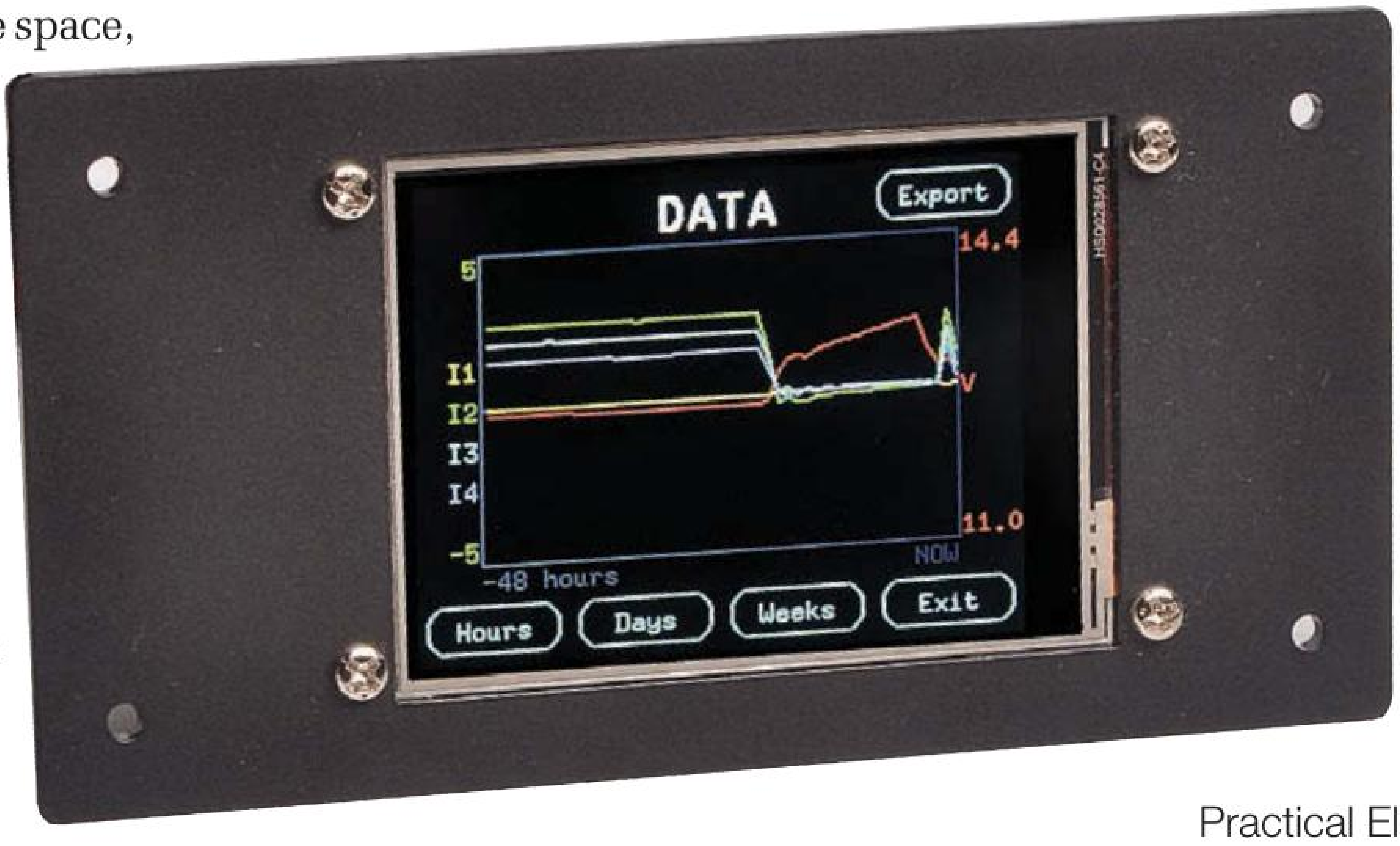
New features

Our design supports up to three shunts, so it can monitor three separate current paths, helping you to split out the charging or discharging figures across multiple loads and/or generators. It even includes a fourth internal shunt for monitoring its own power usage.

For example, you might have a solar panel array and a wind generator (or several) and want to keep track of the energy they generate separately. Or you might have several loads like a fridge, lights and a kettle and want to see which one is consuming the most energy.

The design allows 100V at its input and the PIC32 we have used has plenty of storage space, so it can record more data for long periods.

The battery voltage and currents are sampled at 10-second intervals. That data is averaged every hour to give up to two days of hourly samples. The hourly samples are also averaged over each day to give about a fortnight of daily values.



The flow of both charge and energy is logged, to provide capacity values in Ah (amp-hours) and Wh (watt-hours). You specify the full and empty voltages of your battery, plus the battery capacity, so that the unit can self-calibrate when the battery is either fully charged or discharged.

A simple, linear voltage state-of-charge value is also calculated, giving a rough indication of battery state when the more accurate information is not available.

Operating concept

Fig.1(a) shows the simplest way to use the *Battery Monitor Logger*. The battery connects to a two-way screw terminal (CON3) while the positive ends of up to three loads or charging sources connect to the contacts of three-way screw terminal CON3a.

The negative ends of those loads/charging sources connect directly to the battery negative (ground).

This allows the *Battery Monitor Logger* to independently measure and display the current flowing to or from each load or charging source.

It also produces a total current in/out figure and uses this to keep track of the battery's state of charge in amp-hours (Ah). Multiplying this by the battery's current voltage gives a nominal watt-hours (Wh) figure for the current state of charge.

If you have more than three external devices to connect, they can share terminals on CON3a, as shown in Fig.1(b). For example, one terminal is shared by two loads (LOAD1 and LOAD2). The measurement on that channel will be the total load current for these two devices. Another terminal is shared by two charging sources (SOLAR and WIND), and likewise, their currents will be summed.

The third terminal is shared by LOAD3 and a mains charger. In this case, the unit will measure the net current flow in/out – ie, it will see a flow into the battery if the charger current exceeds the current drawn by LOAD3, a flow out if the situation is reversed, and will measure zero if the two currents are equal (ie, the LOAD3 current is supplied by the charger).

If you need to monitor currents over 10A, you can use the same arrangement except with external shunts. These will typically have a lower resistance and also can handle higher dissipation, both factors allowing greater currents to flow safely. For example, you can get 100A shunts quite easily, or even 500A shunts.

Circuit design

The circuit of the *Battery Monitor Logger* is shown in Fig.2. It has been designed as a complete Micromite-compatible board, rather than an add-on board for a *Micromite LCD BackPack*.

This allows us to control its power usage better, reducing the current drawn from the battery. As with any battery-operated device, it's important to consider power consumption during the design phase.

The battery and load/charger terminals are at lower right, with the bottom half of the right-hand page showing the sensing circuitry. Other external connections (USB, serial, programming etc) are arranged along the left-hand side, with the *BackPack* circuitry occupying most of the left-hand page, plus the display at centre-right. The unit's power supply is across the top of both pages.

The *Micromite V2 BackPack* (May 2018) is the closest *BackPack* variant to our design. This comparison is only for the sake of explaining some of our design choices; it is not important if you are coming to this circuit without knowing about the earlier designs.

We've opted to use the 2.8in (7cm diagonal) LCD touchscreen in this design, rather than the 3.5in (9cm) version we've been using more recently (eg, in the *V3 BackPack*), as the smaller display uses slightly less power.

The *V3 BackPack* also has many features which simply aren't needed in this case, hence our choice of the *V2 BackPack* as the basis for this design. The main advantage it has compared to the original *Micromite BackPack* is the inbuilt USB-Serial interface.

Battery sensing

The main battery sensing circuitry centres on IC5 (an AD7192) and REF1 (a MAX6071). IC5 is a four-channel 24-bit ADC (analogue-to-digital converter) with an SPI serial interface. It is supplied from REG2's 3.3V output, with its analogue rail filtered by a 10µH inductor. Each of its 3.3V supply pins is bypassed by a 100nF capacitor.

IC5 shares the SPI bus with the LCD touchscreen, with IC1's pin 24 used for the $\overline{\text{CS}}$ function, to indicate when IC5 is being addressed.

IC5 needs a stable reference voltage to convert voltages into digital values, and this comes from REF1, a MAX6071 2.5V reference. It is a very low-noise and precise voltage reference chip, and it is supplied with 3.3V from REG2, with 100nF capacitors on its input and output. Its output supplies IC5's REFIN1+ (pin 15), while IC5's REFIN1– (pin 16) is tied to analogue ground.

Each of the four analogue inputs to IC5 is fed by a 390kΩ/10kΩ divider, bypassed at the bottom by a 100µF capacitor. This means that the nominal full-scale reading is 100V with a resolution of around 6µV, and settling times of around ten seconds. We use the ADC to perform a conversion cycle (of all channels) about once every ten seconds, a slow rate needed to obtain maximum resolution.

One of the dividers is connected directly across the battery at CON3. The other three monitor the voltage at the load/charger end of the three shunts which connect between the BAT terminal of CON3 and the terminals of CON3A. By measuring the difference between the voltages fed to the ADC, we can determine the current flow into or out of each terminal.

The PCB provides pads for 15mΩ shunt resistors which allow a theoretical resolution under 10mA. These are 3W

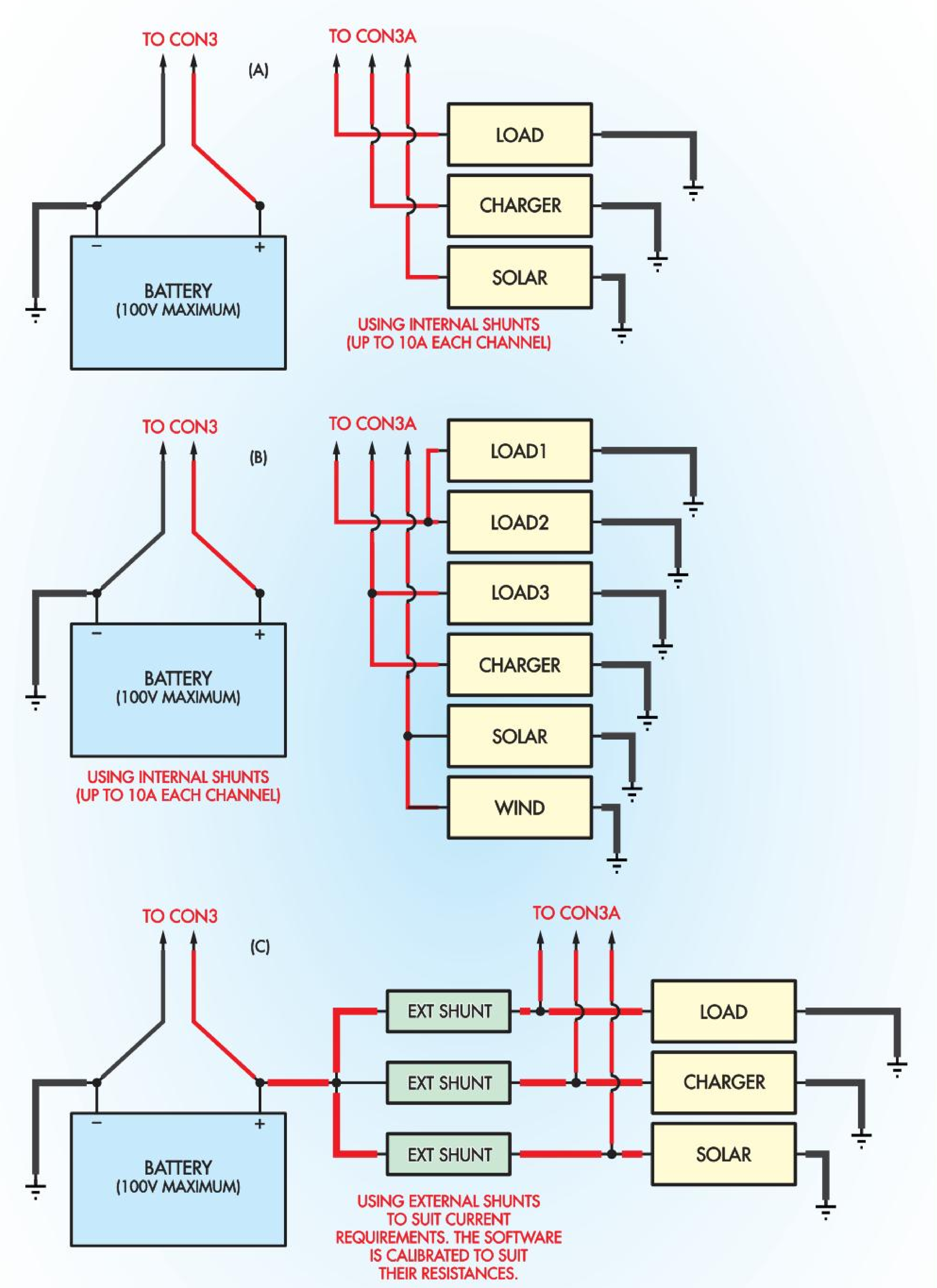


Fig.1: three examples of how you could use the Battery Logger/Monitor. The simplest configuration, at top, uses its internal shunts to monitor the currents (up to 10A) into or out of three loads/charging sources. Or as shown in (B), you can connect more than three loads/charging sources, with some of them sharing shunts. For higher-current applications (up to hundreds of amps), external shunts can be used, as in (C).

parts, notionally allowing up to 14A to be sensed. In practice, the terminals limit this to around 10A.

If larger external shunts are used instead, you just need to run low-current sensing wires from both their ends, back to CON3/CON3A. The shunt values can be set in the software to account for practically any resistance value.

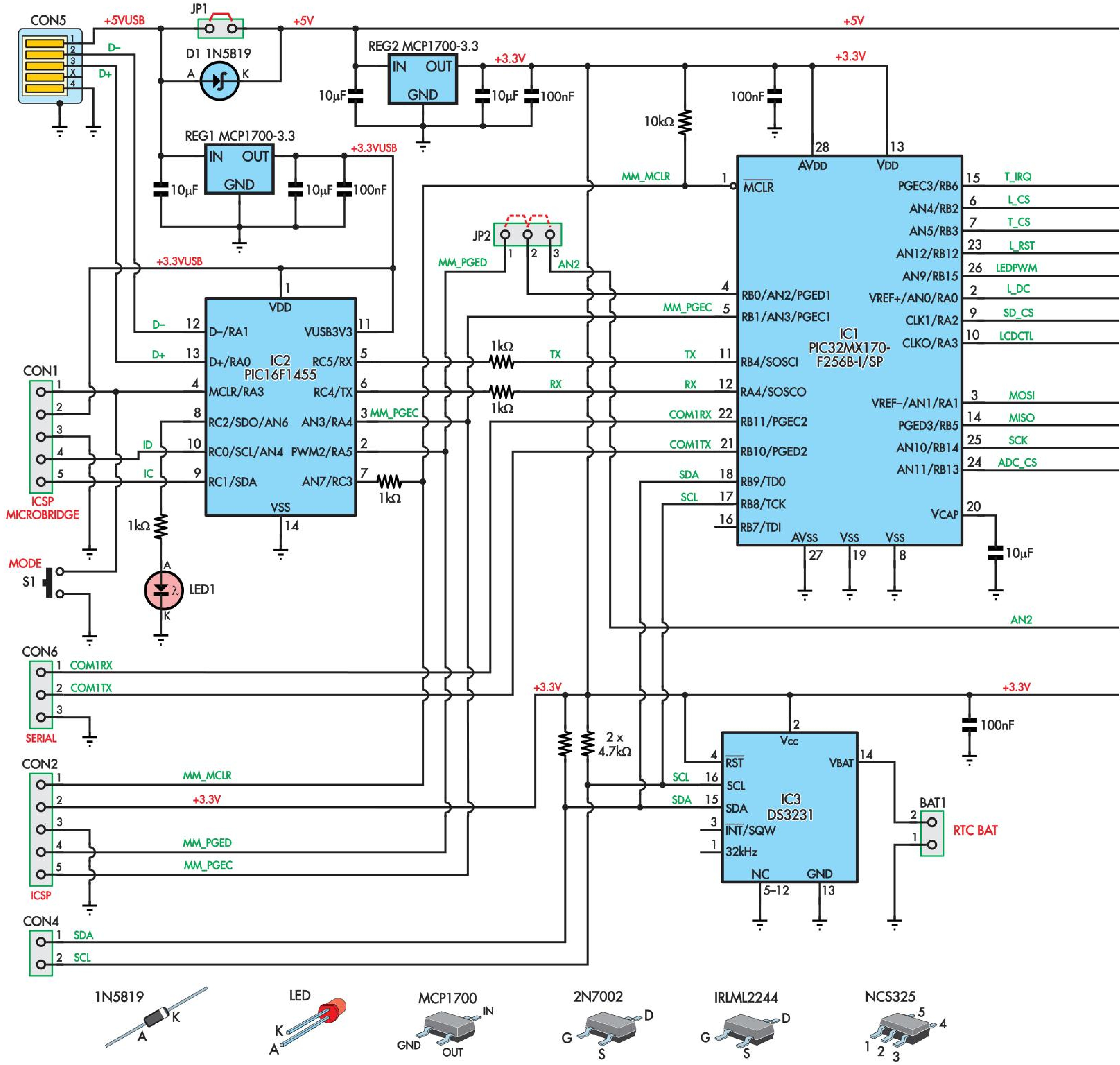
A local analogue ground net separates the analogue voltages from digital SPI signals.

Supply current

The current drawn by the circuit itself is modest but not insignificant, and needs to be accounted for to get accurate measurements. Since it is a fairly low current, we use a different technique to monitor it. Any current flowing into our circuit from the battery at CON3 flows out through a 100mΩ shunt resistor, generating a voltage below ground proportional to the current.

IC6 is a single-channel op amp in a five-pin SOT23-5 SMD package. It is wired as an inverting amplifier with a gain of 100 (100kΩ/1kΩ), presenting a voltage to IC1's pin 4 where the micro's internal ADC can read it. The 100nF capacitor and 100kΩ resistor provide similar smoothing on this signal (a time constant of around ten seconds) so that it too can be sampled at similar intervals to the other channels.

When the Battery Monitor Logger is operating, the LED backlight of the LCD panel consumes the most power, so


CON5
+5VUSB
JP1
+5V
D–
D+
D1 1N5819
A
K
REG2 MCP1700-3.3
IN
OUT
GND
+3.3V
10µF
10µF
100nF
100nF
10kΩ
REG1 MCP1700-3.3
+3.3VUSB
10µF
10µF
100nF
MM_MCLR
JP2
MM_PGED
AN2
+3.3VUSB
VDD
IC2
PIC16F1455
D–/RA1
D+/RA0
MCLR/RA3
RC2/SDO/AN6
RC0/SCL/AN4
RC1/SDA
VUSB3V3
RC5/RX
RC4/TX
AN3/RA4
PWM2/RA5
AN7/RC3
VSS
MM_PGEC
1kΩ
TX
RX
CON1
ICSP
MICROBRIDGE
ID
IC
1kΩ
LED1
MODE
S1
IC1
PIC32MX170-
F256B-I/SP
AVDD
VDD
MCLR
RB0/AN2/PGED1
RB1/AN3/PGEC1
RB4/SOSCI
RA4/SOSCO
RB11/PGEC2
RB10/PGED2
RB9/TDO
RB8/TCK
RB7/TDI
AVSS
VSS
VSS
PGEC3/RB6
AN4/RB2
AN5/RB3
AN12/RB12
AN9/RB15
VREF+/AN0/RA0
CLK1/RA2
CLKO/RA3
VREF–/AN1/RA1
PGED3/RB5
AN10/RB14
AN11/RB13
VCAP
T_IRQ
L_CS
T_CS
L_RST
LEDPWM
L_DC
SD_CS
LCDCTL
MOSI
MISO
SCK
ADC_CS
COM1RX
COM1TX
SDA
SCL
10µF
AN2
CON6
COM1RX
COM1TX
SERIAL
+3.3V
2 x
4.7kΩ
100nF
IC3
DS3231
Vcc
RST
SCL
SDA
INT/SQW
32kHz
NC
GND
VBAT
BAT1
RTC BAT
CON2
MM_MCLR
+3.3V
MM_PGED
MM_PGEC
ICSP
CON4
SDA
SCL
1N5819
LED
MCP1700
IN
GND
OUT
2N7002
G
D
S
IRLML2244
NCS325
1 2 3 4 5

Battery Multi-logger

Fig.2: the circuit includes the equivalent of an entire Micromite V2 BackPack, a precision multi-channel ADC and a switchmode regulator capable of running the device from a DC supply between 6V and 100V. It monitors the battery voltage, the current to/from three external points and its own current consumption and logs all this (plus the current battery state-of-charge) to the internal Flash memory of microcontroller IC1.

a high PWM frequency is used to ensure that this measurement is accurate.

Power supply

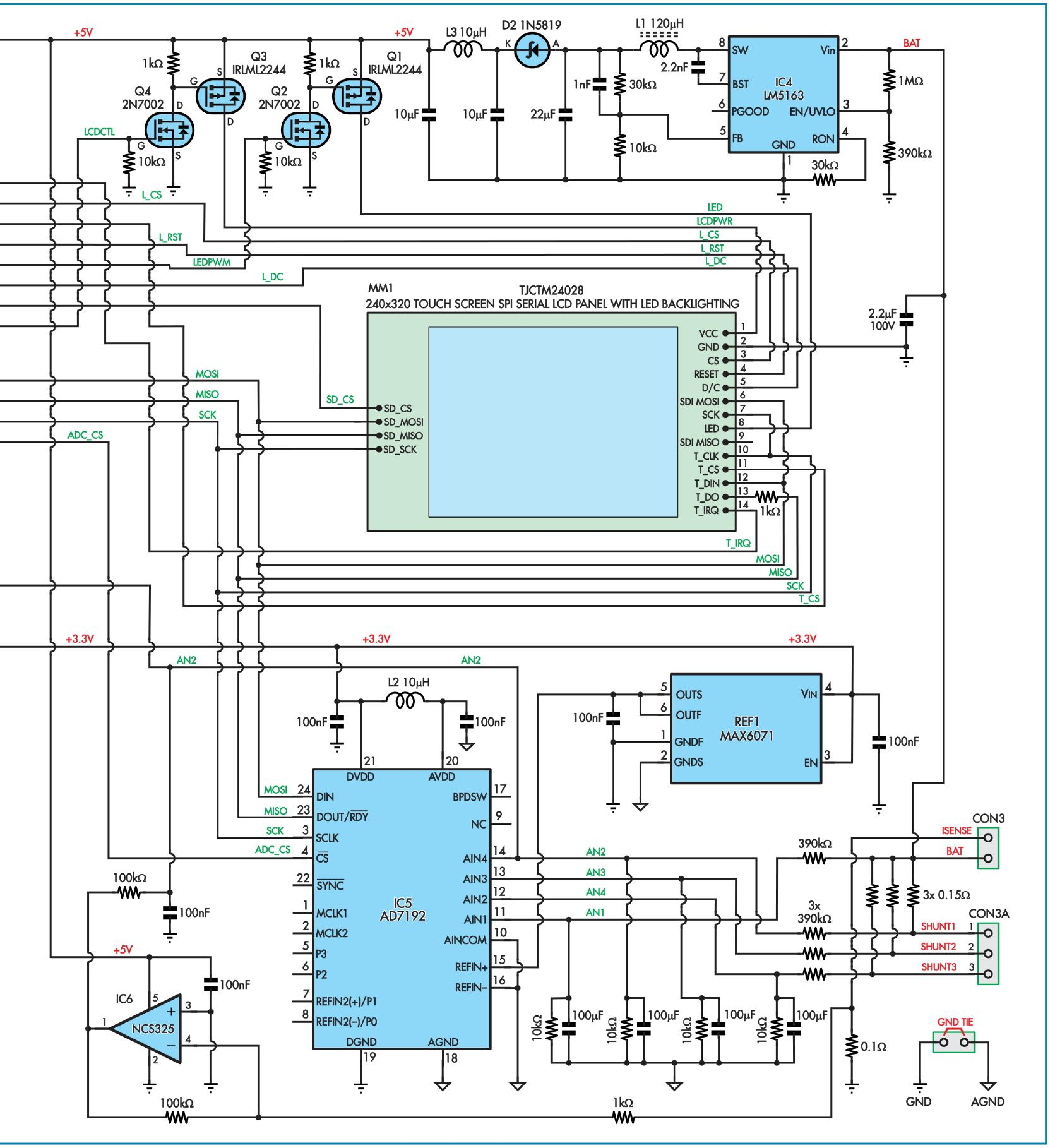
There are two possible power sources in this circuit; USB socket CON5 can supply 5V, while the battery connection at CON3 handles up to 100V from the battery being monitored. (Several components on the board have a 100V maximum rating, so this is a hard limit and should not be exceeded.)

A switchmode buck regulator chip, IC4 (LM5163) efficiently steps the battery voltage down to 5V. Its supply from the battery via CON3 is bypassed with a 2.2µF capacitor and fed into pins 2 (VIN) and 1 (GND).

A voltage above 1.5V on pin 3 (EN) enables the regulator, which is equivalent to a voltage of around 5.5V at CON3 due to the 1MΩ/390kΩ resistive divider.

Apart from accepting up to 100V at its input, IC4 also has an extremely low idle current of just 10.5µA with no load, and not much more at light loads. Its efficiency varies with the input voltage and load current, but is typically in the 75-90% range. See the panel below for more details on this handy little chip.

It switches its pin 8 output (SW) alternately between VIN and GND using a pair of internal N-channel MOSFETs. The upper MOSFET has its gate voltage supplied from the 2.2nF capacitor on pin 7 (BOOST).


+5V
1kΩ
Q3 IRLML2244
1kΩ
Q1 IRLML2244
Q4 2N7002
Q2 2N7002
LCDCTL
10kΩ
10kΩ
L_CS
+5V
L3 10µH
D2 1N5819
L1 120µH
2.2nF
10µF
10µF
22µF
1nF
30kΩ
10kΩ
IC4 LM5163
SW
Vin
BST
PGOOD
EN/UVLO
FB
GND
RON
30kΩ
BAT
1MΩ
390kΩ
LED
LCDPWR
L_CS
L_RST
L_DC
L_RST
LEDPWM
L_DC
MM1
TJCTM24028
240x320 TOUCH SCREEN SPI SERIAL LCD PANEL WITH LED BACKLIGHTING
VCC
GND
CS
RESET
D/C
SDI MOSI
SCK
LED
SDI MISO
T_CLK
T_CS
T_DIN
T_DO
T_IRQ
1kΩ
SD_CS
SD_MOSI
SD_MISO
SD_SCK
SD_CS
MOSI
MISO
SCK
ADC_CS
2.2µF 100V
T_IRQ
MOSI
MISO
SCK
T_CS
+3.3V
+3.3V
+3.3V
AN2
AN2
L2 10µH
100nF
100nF
100nF
REF1 MAX6071
OUTS
OUTF
GNDF
GNDS
VIN
EN
100nF
IC5 AD7192
DVDD
AVDD
DIN
DOUT/RDY
SCLK
CS
SYNC
MCLK1
MCLK2
P3
P2
REFIN2(+)/P1
REFIN2(–)/P0
DGND
AGND
BPDSW
NC
AIN4
AIN3
AIN2
AIN1
AINCOM
REFIN+
REFIN–
MOSI
MISO
SCK
ADC_CS
100kΩ
100nF
+5V
100nF
IC6
NCS325
100kΩ
1kΩ
AN2
AN3
AN4
AN1
390kΩ
3x 390kΩ
3x 0.15Ω
10kΩ
100µF
10kΩ
100µF
10kΩ
100µF
10kΩ
100µF
0.1Ω
CON3
ISENSE
BAT
CON3A
SHUNT1
SHUNT2
SHUNT3
GND TIE
GND
AGND

Features and specifications

- Battery voltage: 6-100V
- Current monitoring: up to three chargers or loads, monitored separately
- Current handling: limited only by the shunts used (10A with onboard shunts)
- Current resolution: 0.1% (10mA with onboard shunts)
- Operating current: <1mA while logging (with display off)
- User interface: 2.8-inch colour touchscreen
- Firmware: Programmed in BASIC
- Data logging: can be viewed on device graphically, or downloaded as CSV files
- Measurements: current charge (Ah) and energy (Wh)
- State of charge: displayed based on voltage and charge.

The pulses are smoothed by the 120µH inductor and a 22µF capacitor to provide the output voltage. The voltage on feedback pin 5 (FB) is internally compared to a 1.2V reference, so the 30kΩ/10kΩ divider sets the output voltage to 4.8V.

This is set to be slightly less than 5V so that if an alternative 5V supply is available, it takes over from the battery. Schottky diode D2 feeds the 4.8V into a pi filter formed of two further 10µF capacitors and a 10µH inductor.

The 1nF capacitor across the 30kΩ resistor at the top of the FB divider helps with the stability of the circuit that drives the output pulses, by ensuring sufficient ripple at the FB pin for the circuit to operate correctly. See our panel for more detail on this.

Microcontroller details

This approximately 5V rail then feeds the Micromite section of the circuit. MCP1700-3.3 REG2 and its associated bypass capacitors provide the 3.3V supply for microcontroller IC1. This is a 32-bit, 50MHz micro (PIC32MX170F256B) and is surrounded by its own complement of bypass capacitors.

IC1 is programmed with the MMBasic firmware and runs a BASIC program to implement the *Battery Monitor Logger* functions.

While some *Micromite BackPacks* used the 28-pin DIP version of this IC, the *Battery Monitor Logger* uses the 28-pin SMD (SOIC) part. It works identically but is smaller, so we can cram more onto the PCB, and most of the other ICs are only available as SMDs anyway. In this case, its pins are relatively far apart (on a 1.27mm/0.05in pitch) so it is not difficult to solder.

To save power, the micro can switch 5V power on and off to the touchscreen via the 14-way LCD header. A high level on IC1's pin 10 turns on N-channel MOSFET Q4, which is otherwise held off by a 10kΩ pull-down resistor. When Q4 is on, it pulls P-channel MOSFET Q3's gate low, which allows 5V to flow from Q3's source to drain and into the LCD panel's supply pin.

A similar arrangement, controlled by IC1's pin 26 via MOSFETs Q2 and Q1, switches power to the LCD panel's LED backlight. Typically, a PWM signal is applied to pin 26, modulating the backlight brightness.

Unlike the *Micromite BackPack V2*, which had PWM brightness control, we have omitted the option of manual backlight control as the backlight is easily the biggest user of power in the circuit. So it needs to be fully shut off during logging and monitoring.

Serial communications

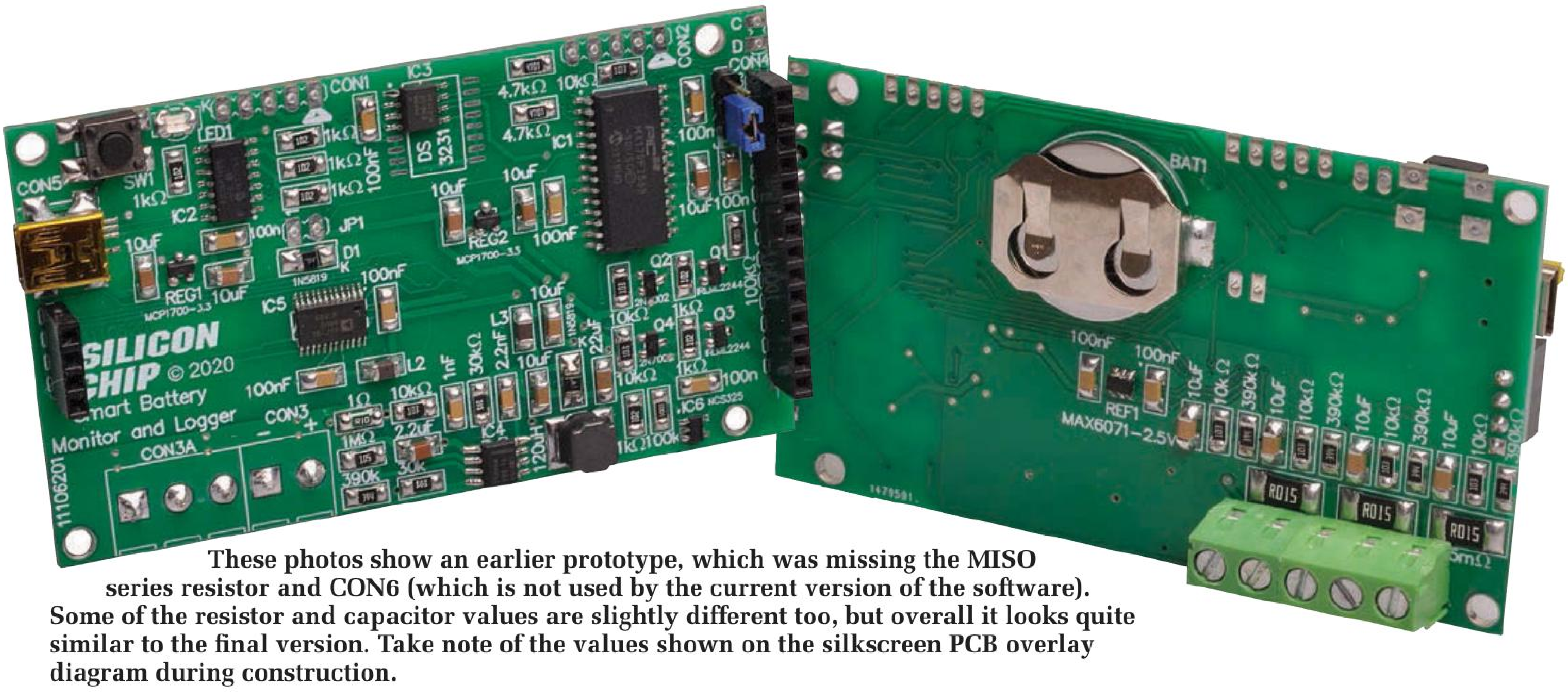
IC1 sends display data and gets touch events back from the touchscreen using an SPI serial bus on its pins 3, 14 and 25 (MOSI, MISO and SCK). These connect to the LCD panel's pin 6 and 12 (MOSI), pin 13 (MISO) and pins 7 and 10 (SCK). 'MISO' stands for 'master in, slave out' while 'MOSI' stands for 'master out, slave in'.

The MISO line has a series 1kΩ resistor so that it can still operate when the LCD panel is switched off. These signals, plus a chip select signal from IC1's pin 9, also connect to the SD card header at the other end of the LCD panel PCB via a four-pin header.

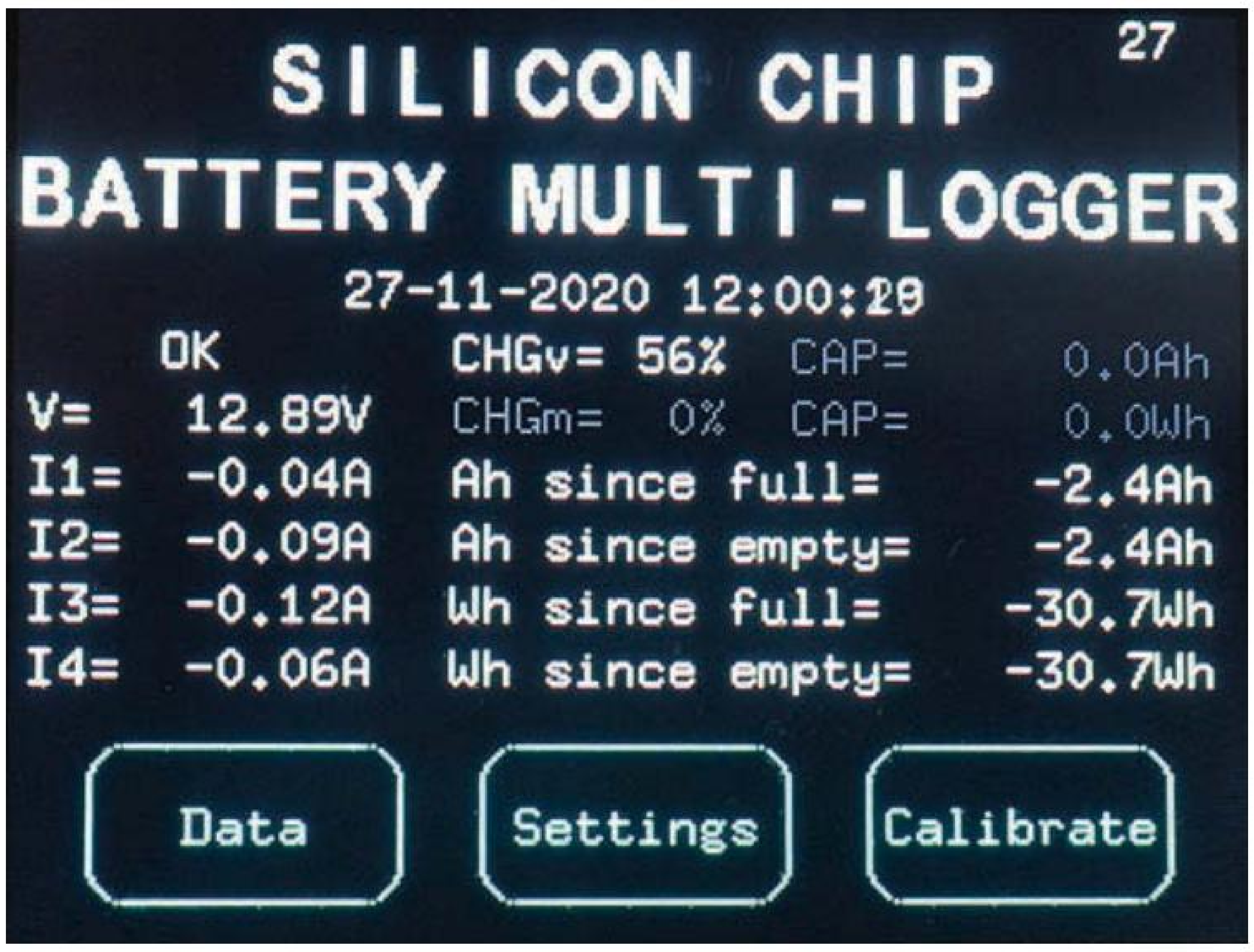
We had planned to use the SD card to store data, but Flash memory limitations in the micro mean that there isn't enough space to include the (rather large) libraries needed to do this.

IC2 is an 8-bit PIC16F1455 microcontroller programmed with the Microbridge firmware. This allows it to act as a USB-Serial bridge, and it can also be used to program the PIC32 microcontroller.

Pushbutton S1 is used to switch IC2 between USB-Serial and programming modes, with LED1 flashing to indicate that it is passing serial data, or lighting up solidly when in programming mode.



These photos show an earlier prototype, which was missing the MISO series resistor and CON6 (which is not used by the current version of the software). Some of the resistor and capacitor values are slightly different too, but overall it looks quite similar to the final version. Take note of the values shown on the silkscreen PCB overlay diagram during construction.



27

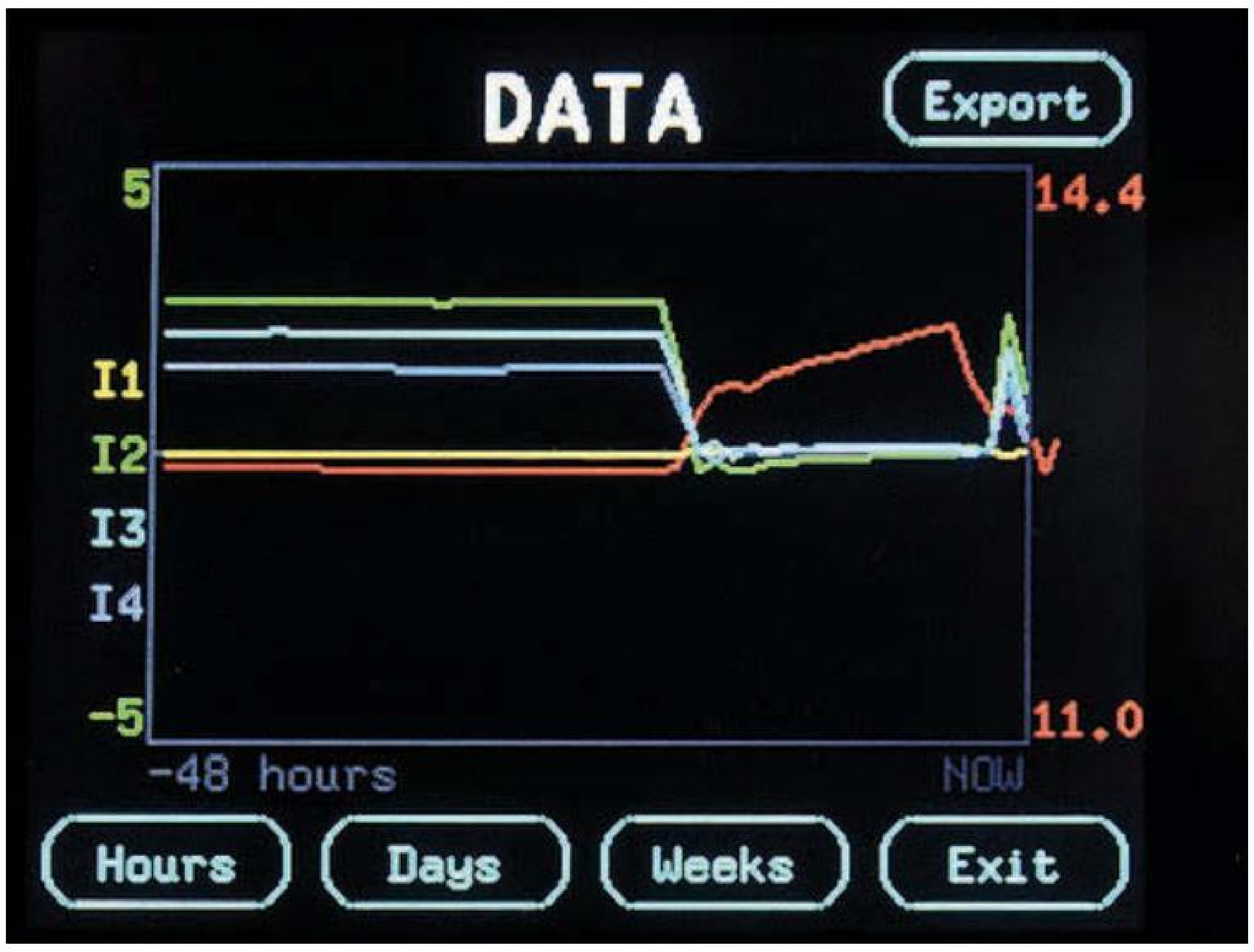
SILICON CHIP

BATTERY MULTI-LOGGER

27-11-2020 12:00:29

	OK	CHGv= 56% CAP=	0.0Ah
V=	12.89V	CHGm= 0% CAP=	0.0Wh
I1=	-0.04A	Ah since full=	-2.4Ah
I2=	-0.09A	Ah since empty=	-2.4Ah
I3=	-0.12A	Wh since full=	-30.7Wh
I4=	-0.06A	Wh since empty=	-30.7Wh

Screen1: The main screen provides all the critical statistics for your battery, as well as three simple menu options for accessing other features. The greyed values seen are capacity calculations which are not yet valid, as the Logger has not detected a complete charge and discharge cycle; they will light up brighter when that happens.



Screen2: The Data screen provides a graphical view of the logged data. Different timespans can be shown, and the display will automatically scroll once a minute to show current data. The Weeks option provides around a fortnight of data. Data can also be dumped as CSV rows over the console serial port with the Export button.

Mini USB Type-B socket CON5 is used both for USB communications (D+/D–) as well as optionally supplying 5V power. Schottky diode D1 feeds USB 5V to the Micromite 5V rail. Jumper JP1 provides the means to bypass D1 if needed.

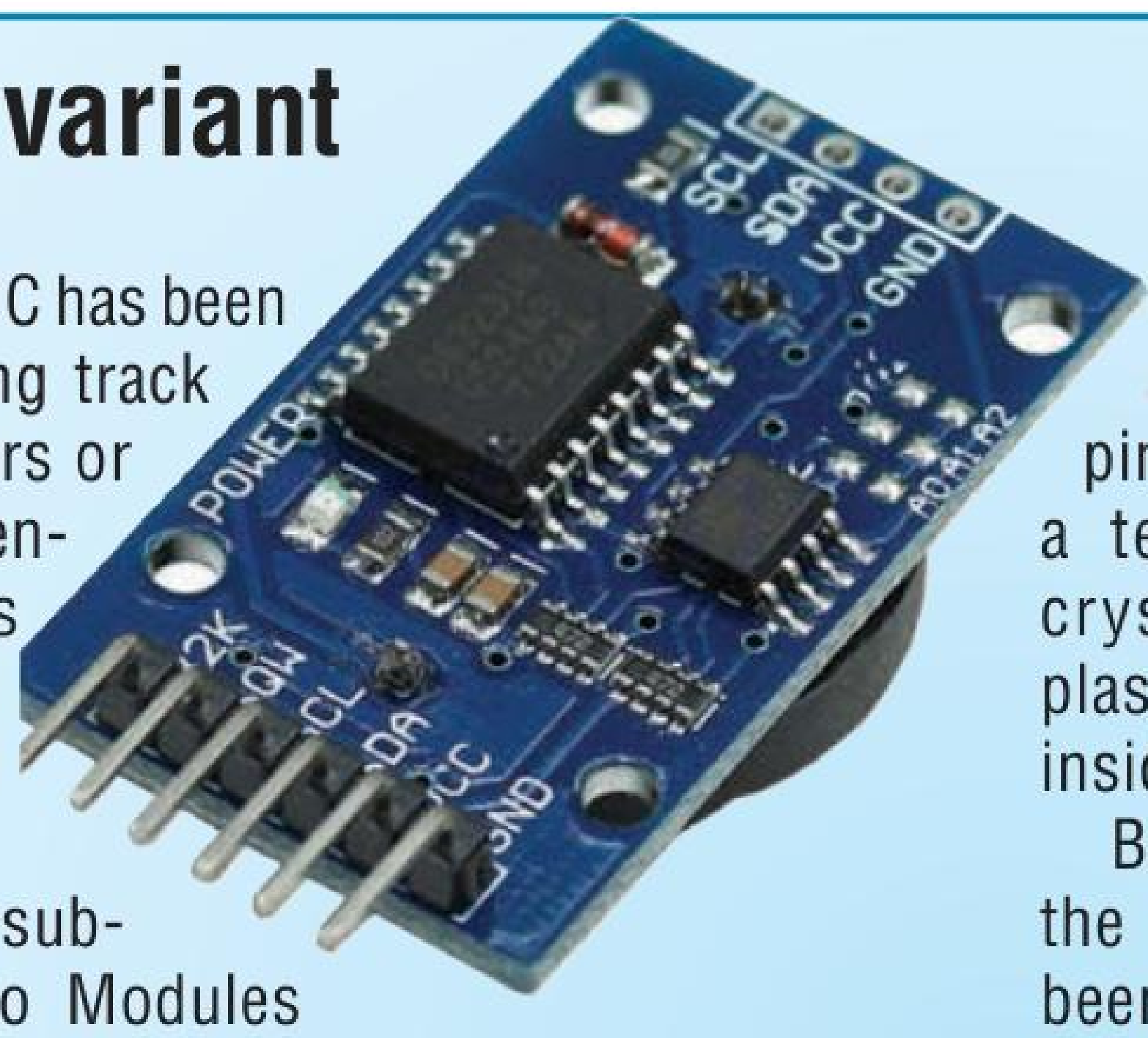
REG1 is identical to REG2 and supplies 3.3V to IC2 independently. Serial TX and RX signals are bridged to and from the virtual USB-Serial port by IC2. These connect between its pins 5 and 6, via 1kΩ resistors, to Micromite console pins 11 and 12 on IC1.

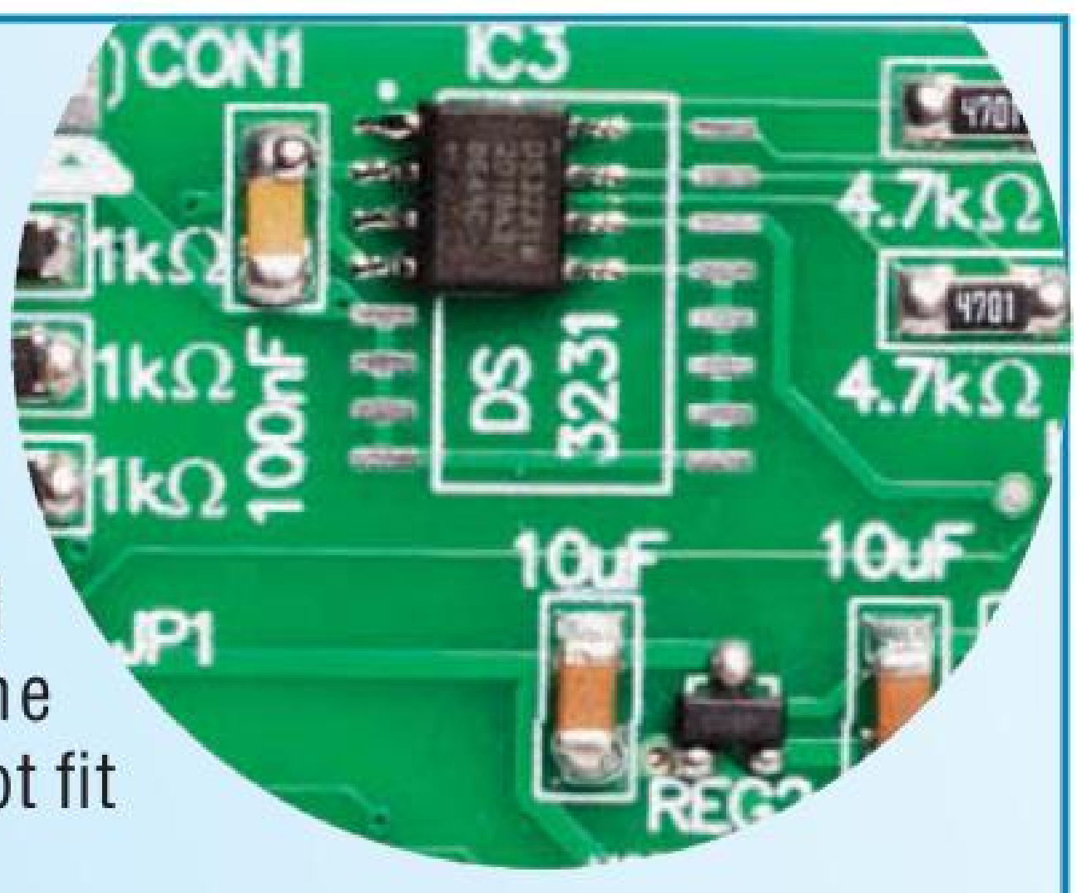
IC2's pins 2, 3 and 7 can be used to program IC1 via its ICSP interface; they are connected to IC1's pins 4, 5 and 1 respectively. The PGD signal travels via JP2, which allows IC1's pin 4 to be used as an analogue input when it is not being used for programming.

Both IC1 and IC2 have their in-circuit serial programming (ICSP) pins broken out to the edge of the PCB at CON2 and CON1 respectively. This is a feature not seen on the other *BackPacks*, but we have included it here because the SMD ICs used here are more difficult to program out-of-circuit than through-hole (DIP) chips.

A DS3231 real-time clock (IC3) provides accurate timekeeping over long periods. Its I^{2}C serial bus pins 15 and 16 (SDA and SCL) connect to IC1 at pins 18 and 17, the I^{2}C pins used by the Micromite firmware. Two 4.7kΩ resistors provide the pullups needed by the I^{2}C protocol.

DS3231 MEMS variant





The DS3231 real-time clock IC has been the go-to choice for keeping track of time for the last five years or so. Its appeal is no doubt enhanced by the fact that it is available in an easy-to-use module typically sold as an Arduino accessory.

Such a module was the subject of our first El Cheapo Modules feature from January 2018, which we used in several projects, typically in combination with a Micromite. The module includes I^{2}C pullup resistors, an I^{2}C EEPROM and a cell holder.

The module simplifies connection as it includes all that is needed for the DS3231 chip to work, but sometimes it's too big. We used the bare DS3231 IC (which comes in a wide 16-pin SOIC SMD package) in our Micromite BackPack V3 (August 2020) and the Ol' Timer II clock (August 2021).

To support those projects, we kept a stock of those ICs. One day, we were surprised to receive a package of small 8-pin SOIC parts instead of the wide 16-pin SOICs that we were expecting. Had we been conned?

No; we had received the DS3231M variant instead. Those familiar with the DS3231 will know that it only uses eight of its pins; the lower pins are marked NC ('not connected'). The reason for the large package is not that it needs 16 pins, but because it includes a temperature-compensated crystal oscillator inside the plastic IC case, which would not fit inside an 8-pin package chip.

But with the advance of MEMS technology, the crystal oscillator inside the DS3231 has been superseded by a smaller MEMS device.

So given their small size and decent performance, we decided to try them out in this project. We found the DS3231M to work the same as the DS3231. The nominal accuracy is slightly worse at ±5ppm compared to ±3.5ppm, but for situations where size is of concern, the smaller package is the overriding concern.

The MEMS part doesn't appear to suffer from crystal ageing either, which means that in the longer term, it could be more accurate unless this is compensated for in the earlier version of the chip. The backup battery current draw appears to be higher for the MEMS part in typical cases, but in most cases, the battery life will still be close to its shelf life.

In this particular project, we've made allowances for either part in the PCB design, with a dual footprint that suits both the wide 16-pin SOIC part and the narrower 8-pin SOIC part. We don't know if the DS3231M will end up more popular than the original DS3231, but we're ready for either eventuality.

– Parts list – Battery Monitor Logger

1 double-sided PCB coded 11106201, measuring 86mm x 50mm, available from the PE PCB Service
1 2.8in LCD touch panel with ILI9341 controller
1 UB3 Jiffy box (optional, depending on desired mounting)
1 laser-cut acrylic panel to suit LCD and UB3 box
2 5-pin right-angle headers (CON1, CON2; both optional, for programming IC2 and IC1)
1 2-way 5/5.08mm-pitch screw terminal (CON3)
1 3-way 5/5.08mm-pitch screw terminal (CON3A)
2 2-pin headers (CON4 and JP1; both optional)
1 SMD mini-USB socket (CON5)
1 3-way pin header (CON6, serial port; optional)
1 3-pin header (JP2)
2 jumpers/shorting blocks (JP1,JP2)
1 SMD coin cell holder (BAT1) [BAT-HLD-001 – Digi-key, Mouser]
1 CR2032/CR2025 cell or similar (BAT1)
1 120µH 6mm x 6mm SMD inductor (L1) [eg, SRN6045TA-121M – Digi-Key, Mouser etc]
2 10µH 1206/3216-size SMD chip inductors (L2,L3)
1 SMD or through-hole 4-pin tactile pushbutton switch (S1)
1 14-pin header socket strip (for LCD)
1 4-way female socket strip (for LCD)
8 M3 x 6mm panhead machine screws
4 M3 x 12mm tapped spacers
4 M3 x 1mm untapped spacers (eg, stacks of 3mm ID washers)
3 heavy-duty current shunts [eg, Jaycar QP5415, Altronics Q0480 – optional, see text]
hookup / heavy-duty wiring to suit shunts, batteries, load (see text)

Semiconductors
1 PIC32MX170F256B-I/SO 32-bit microcontroller programmed with MMBasic or 11110620A.hex, SOIC-28 (IC1)
1 PIC16F1455-I/SL 8-bit microcontroller programmed with Microbridge firmware,
SOIC-14 (IC2)
1 DS3231/DS3231M real-time clock IC, wide SOIC-16 or SOIC-8 (IC3)
1 LM5163DDAR synchronous buck regulator, SOIC-8 (IC4)
1 AD7192BRUZ 24-bit ADC, TSSOP-24 (IC5)
1 NCS325 CMOS op amp, SOT-23-5 (IC6)
1 MAX6071AAUT25+TT high-precision 2.5V reference, SOT23-6 (REF1)
2 MCP1700-3.3 low-dropout 3.3V regulators, SOT-23 (REG1,REG2)
2 IRLML2244TRPBF P-channel MOSFETs, SOT-23 (Q1,Q3)
2 2N7002 N-channel MOSFETs, SOT-23 (Q2,Q4)
1 3mm or SMD M3216/1206 LED (LED1)
2 SS14 (or equivalent) 40V 1A SMD schottky diodes, DO-214AC (D1,D2)

Capacitors (all SMD M3216/1206 size)
4 100µF 6.3V X5R
7 10µF 50V X7R
10 100nF 50V X7R
1 1nF 50V C0G/NP0
1 22µF 16V X5R
1 2.2µF 100V X7R
1 2.2nF 50V C0G/NP0

Resistors (1% SMD M3216/1206 size 1/8W metal film except where noted)
1 1MΩ (code 105 or 1004)
2 100kΩ (code 104 or 1003)
8 10kΩ (code 103 or 1002)
8 1kΩ (code 102 or 1001)
5 390kΩ (code 394 or 3903)
2 30kΩ (code 303of 3002)
2 4.7kΩ (code 472 or 4701)
1 0.1Ω (code R100 or 0R10)
3 15mΩ 1% 3W (M6331/2512 size; not needed if external current shunts are used)

The PCB is also fitted with a SOIC-8 footprint to allow the similar DS3231M (which uses a MEMS oscillator rather than a crystal) to be used instead. See the separate panel explaining the differences.

Software operation

Some of the following may seem obscure to those not familiar with MMBasic, but this information could come in handy if you want to change the code.

MMBasic certainly makes driving the LCD (TFT) panel easy, as it performs startup initialisation and has built-in BASIC commands for drawing on and writing to the display. But it needs some help to work with our circuit arrangement, which starts with the LCD panel powered off, and therefore not ready to accept the initialisation commands that are automatically sent.

So we need to add a routine (in the `MM.STARTUP` subroutine) to set pin 10 as an output and set it high, then re-run the LCD initialisation code. Every time we power up the display after shutting it down, we need to trigger that code.

We also need to control the other lines that run to the LCD panel, as some of these idle high by default and would therefore waste power. MMBasic does not allow direct control of these, as the firmware reserves them to control the LCD panel, so we need to `POKE` directly to IC1's registers and then run a command to reinitialise the LCD controller.

Similarly, shutting down the controller requires direct POKEs to shut down those pins. No software deinitialisation is needed as the LCD can simply be powered down from any state.

Despite this complication, it's relatively easy to sense touches on the LCD panel even if it is shut down. This is necessary, as the user needs some way to wake the unit up if it is in a low-power state.

Even when the LCD is powered off, the TIRQ pin (which is connected to IC1's pin 15) is pulled to GND whenever the panel is touched. As the Micromite firmware provides a weak pullup on this pin, simply monitoring the state of this pin is sufficient to know if a touch has occurred.

The main job of the MMBasic program is to read the battery voltage and the voltage across the three shunts to infer battery voltages and currents. It logs these to variables which are kept in RAM and they are regularly saved to internal Flash memory.

With the circuit running from the battery it is monitoring, it would take a major fault to shut it down and lose the contents in RAM, so only longer-term samples are saved to Flash memory hourly. If the unit needs to be disconnected to work on the battery, at most one hour of data will be lost.

When saving to Flash, the data is averaged over a period before being archived. This means that less data needs to be stored, but a good amount of data can be kept for historical purposes.

For example, you might like to compare how much power your solar panels are putting into your battery over a period of a few weeks. Data about current and power usage is also used to calculate parameters such as battery capacity and state of charge.

The MMBasic program also provides a user interface to allow settings to be changed and values to be graphed and viewed. Plus there is the option to dump the data over a serial port so that it can be exported to a PC program for graphing and analysis. We'll delve further into the software operation during the setup procedure in *Part 2* next month.

The LM5163 switchmode regulator IC

Our initial design plans for the Battery Logger set the ambitious target of designing it to work at up to 80V, improving on the 60V limit of an older design. That one used an LM2574HV integrated switchmode IC operating at a fixed frequency of 50kHz, requiring a sizeable toroidal inductor and electrolytic capacitor.

Hoping that that state of the art had progressed in the last decade, we decided to look for newer parts. We found plenty of parts capable of working with a 100V supply, which is impressive.

1MHz switching frequencies are no longer uncommon. This much higher switching frequency means that a smaller inductor and capacitors are needed, helping us to keep our board compact.

Many parts we found could only deliver 100mA. While this might have been sufficient with careful control of the LCD backlighting, we wanted more headroom. The LM5163 came in as the cheapest part capable of more than 100mA (500mA) in an easily-soldered SOIC-8 package, which is a good compromise between size and ease of handling.

As is typical of modern buck regulator designs, it is a synchronous type, meaning it has two internal switches. The incoming voltage is switched to the inductor by a high-side internal MOSFET. When the MOSFET is off, a second, low-side MOSFET is switched on to provide a path for the inductor current to circulate. This removes the need for an external diode to serve this role and increases its efficiency.

The LM5163 is a COT (constant on-time) design; the time that the high-side MOSFET is switched on is set by an external resistor, after which it is switched off. The feedback pin monitors the output voltage, and when the output voltage has decayed, another on-cycle begins.

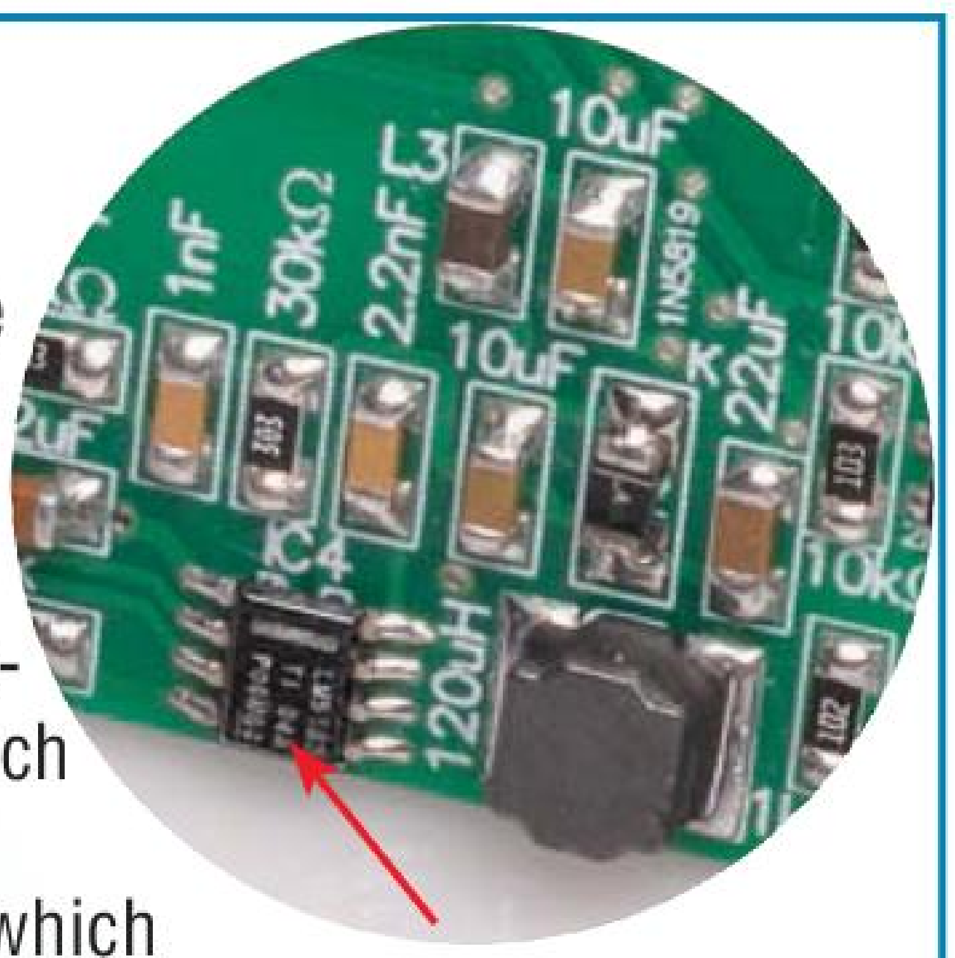
So the duty cycle is modulated to maintain the desired output voltage, but the constant on-time means that the switching frequency varies, although it can be predicted.

When we built our first prototype, everything worked as expected; we were truly impressed with how flexible and easy-to-use this tiny part was. But then, it started squealing! The tone would change with load (which we could easily modulate by adjusting the LCD backlight intensity) and input voltage. It was bad enough, especially around 12V, that we needed to do something about it.

The cause was electrical noise, which was affecting when it would switch on. It might switch on early, which causes the output voltage to rise. This will cause the next switch-on to be delayed, as the controller will be waiting for the output voltage to drop below its threshold.

The output pulses start to cluster into bursts, and it is these clusters that occur at audible frequencies, causing the high-pitched squealing we were hearing ('subharmonic oscillation') – see below.

As we found with our Switchmode 78xx replacement (August 2021), trying to get these sort of parts to operate optimally over a wide range of input voltages can be tricky. In that case, extra output capacitance helped.



Fortunately, a section of the LM5163 data sheet (reproduced in Fig.4) describes methods to avoid this. The aim is to increase the ripple seen by the FB pin, so that the regulator has a clearly defined time to switch on, despite the presence of noise.

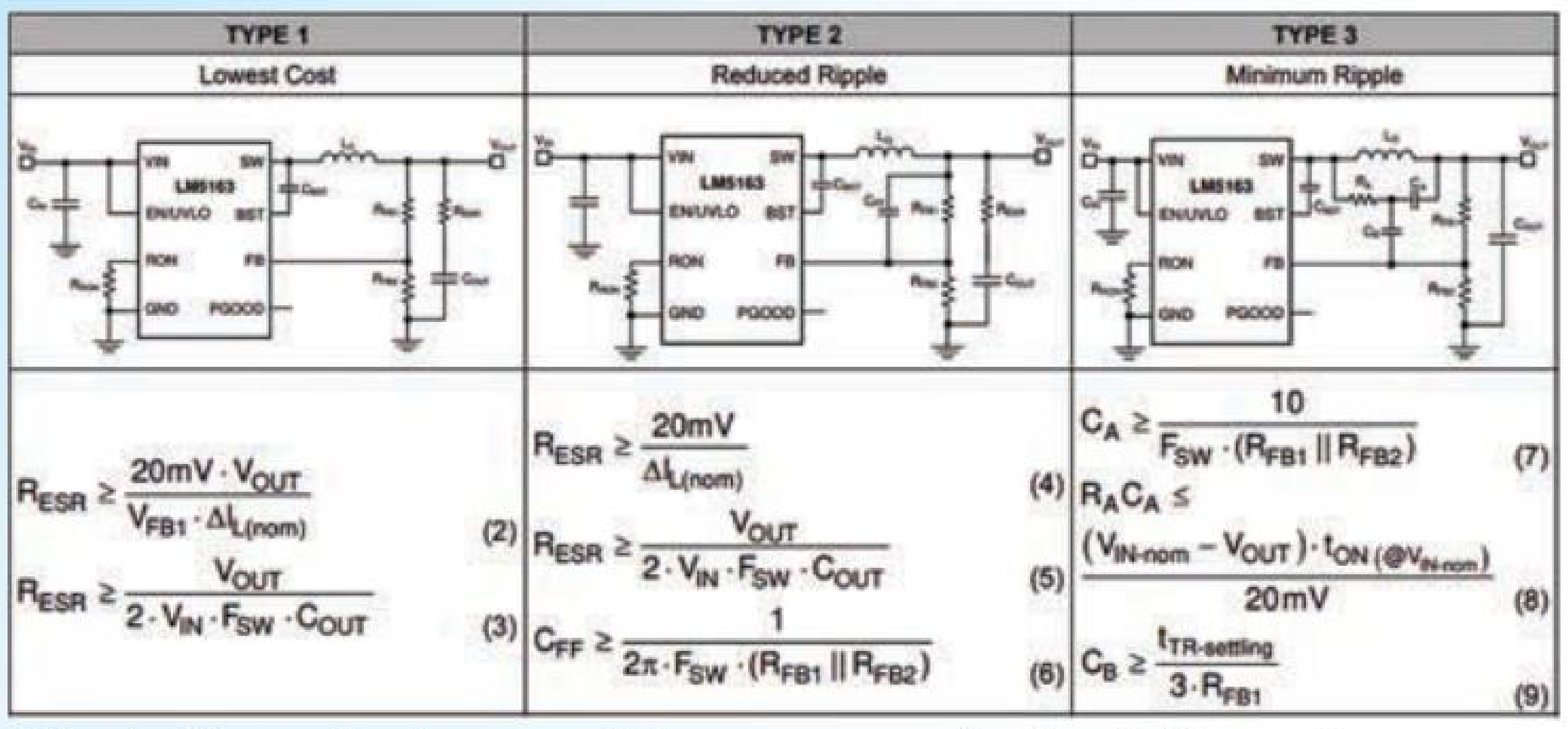
We tried the Type 1 method, which involves adding series resistance to the output capacitor. The extra resistance means that the voltage seen at the FB pin is influenced less by the capacitor and more by the pulses from the inductor. But it also means that the output capacitor is less effective at filtering the output voltage, and we found it did little to reduce the squealing.

So we tried part of the Type 2 method (omitting the series resistor from Type 1) and simply added the 'feedforward' capacitor in parallel with the top feedback divider resistor. This means that the FB pin sees the full amplitude of the output ripple voltage, as it is coupled directly by the capacitor rather than being simply divided by the resistor chain.

This effectively quadruples the ripple seen by the FB pin with our 30kΩ/10kΩ divider, without degrading filtering. That eliminated the squealing, so we have kept it in our final design.

Any switching device which depends on a feedback voltage from a divider to switch its output elements can benefit from having a feedforward capacitor. It depends on the frequency of operation, capacitor value and divider ratio, though.

A word of caution: while this capacitor may appear to be a cure-all, it does have the side-effect of slowing down response to transients as it reduces the closed-loop gain for higher frequency components.


VIN
SW
LM5163
EN/UVLO
BST
RON
FB
GND
PGOOD

TYPE 1	TYPE 2	TYPE 3
Lowest Cost	Reduced Ripple	Minimum Ripple
$R_{ESR} \geq \frac{20mV \cdot V_{OUT}}{V_{FB1} \cdot \Delta I_{L(nom)}}$ (2) $R_{ESR} \geq \frac{V_{OUT}}{2 \cdot V_{IN} \cdot F_{SW} \cdot C_{OUT}}$ (3)	$R_{ESR} \geq \frac{20mV}{\Delta I_{L(nom)}}$ (4) $R_{ESR} \geq \frac{V_{OUT}}{2 \cdot V_{IN} \cdot F_{SW} \cdot C_{OUT}}$ (5) $C_{FF} \geq \frac{1}{2\pi \cdot F_{SW} \cdot (R_{FB1} \parallel R_{FB2})}$ (6)	$C_A \geq \frac{10}{F_{SW} \cdot (R_{FB1} \parallel R_{FB2})}$ (7) $R_A C_A \leq \frac{(V_{IN-nom} - V_{OUT}) \cdot t_{ON(@V_{IN-nom})}}{20mV}$ (8) $C_B \geq \frac{t_{TR-settling}}{3 \cdot R_{FB1}}$ (9)

Fig.4: Texas Instruments' recommended solutions for subharmonic oscillation or 'squegging' in the LM5163. We tried Type 1, and it didn't work, but Type 2 did. It only requires the addition of a low-value feedforward capacitor, Cff, across the upper half of the feedback divider. Type 3 is similar but adds another pole for improved transient response; that's overkill in our application.

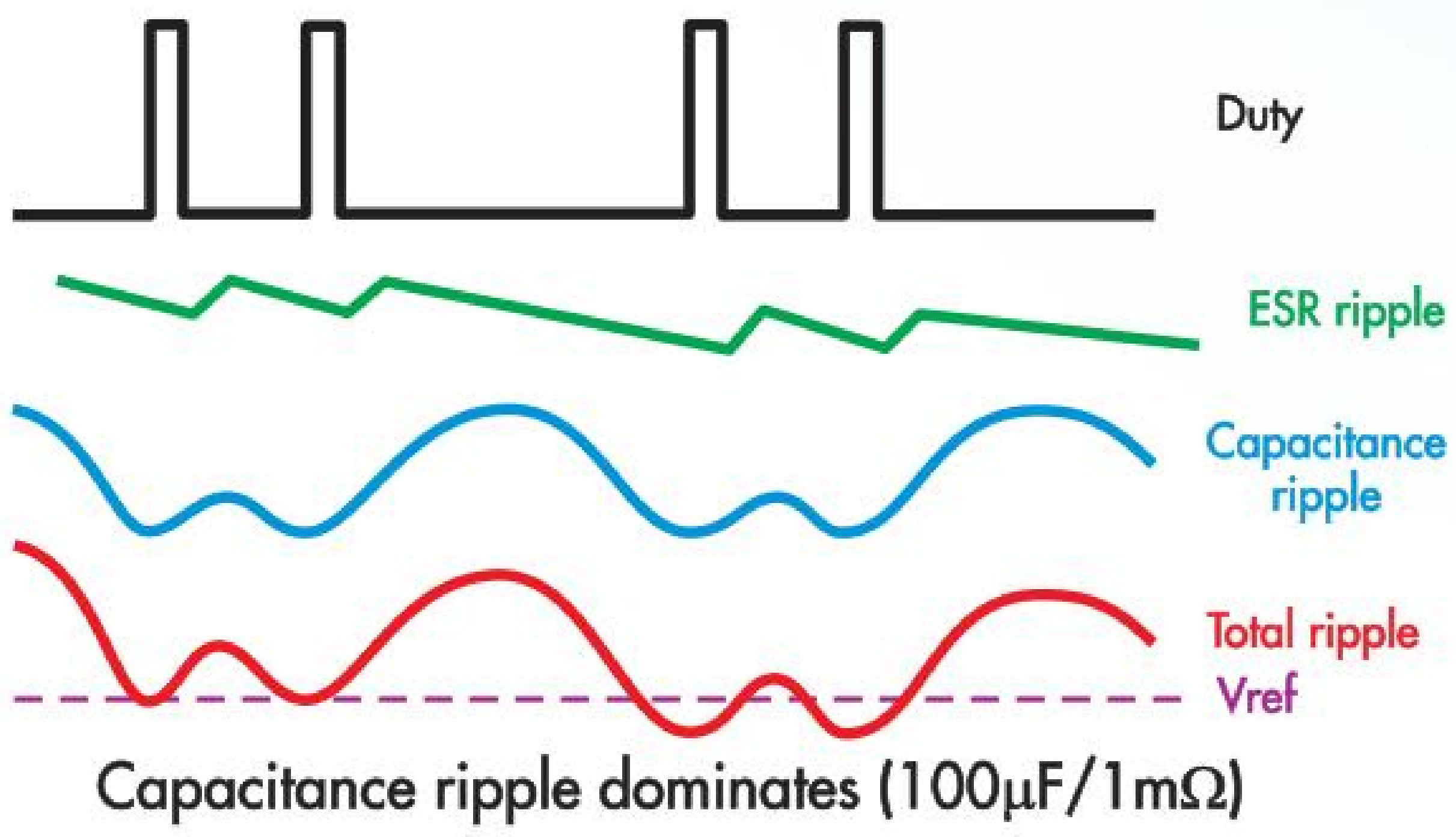
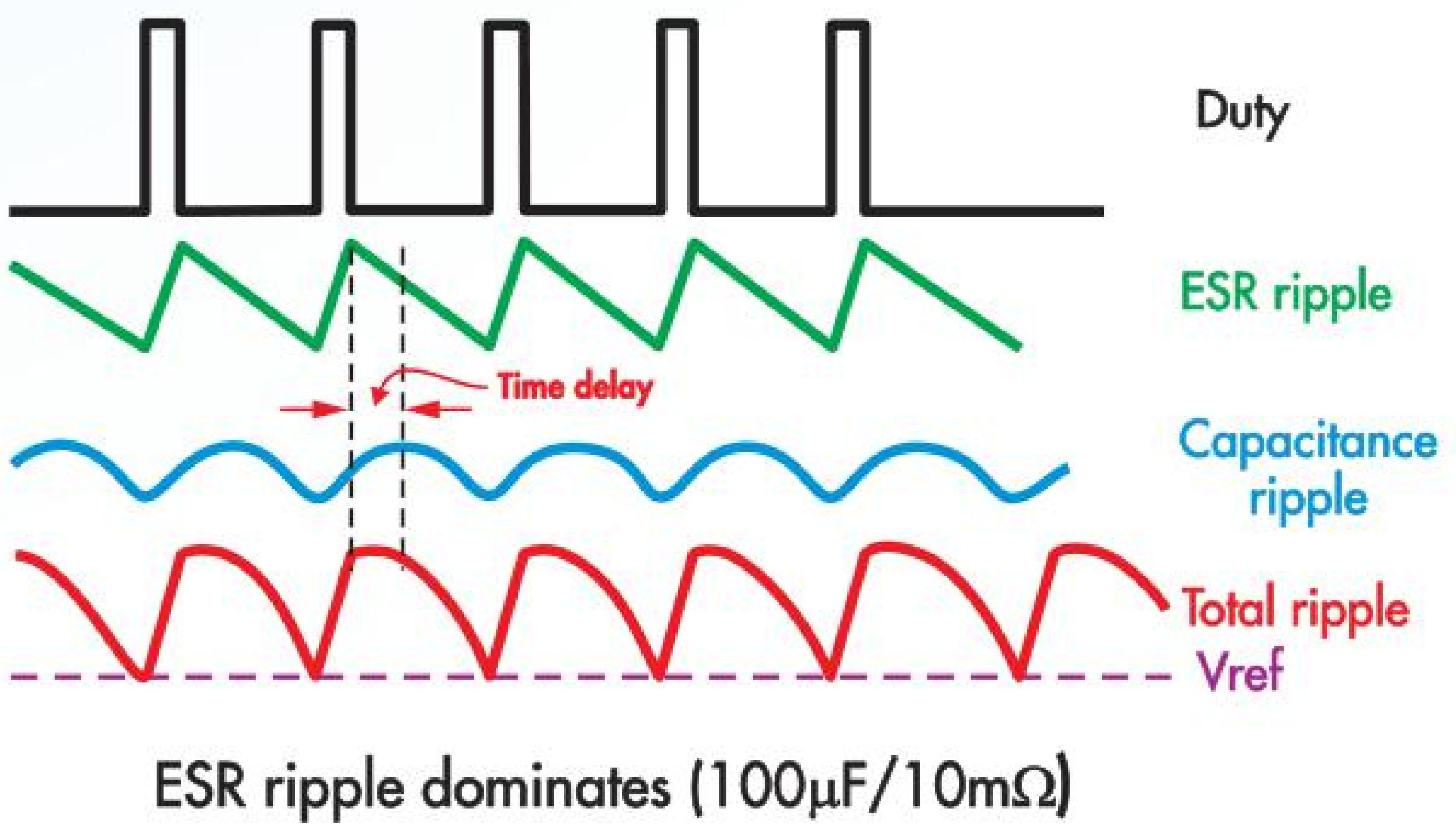


Fig.3: usually, low ESR is considered desirable in a capacitor as it gives superior filtering, but when it filters out the ripple too effectively, it affects the regulator's ability to produce pulses regularly.



Next month

The second and final part of this feature will have the complete PCB assembly details, microcontroller programming procedures, setup and operation instructions, calibration information along with the final construction procedure.

Reproduced by arrangement with
SILICON CHIP magazine 2022.
www.siliconchip.com.au

By John Clarke

ELECTRONIC Wind Chimes

Aaaah . . . wind chimes! They're so soothing . . . listening to the random notes as the wind creates its own melodies. But what do you do if there's no wind? Aim a fan at it? We have a better idea: our *Electronic Wind Chimes* removes your reliance on the wind, and even gives you the possibility of playing tunes using the wind chime, enriching the experience!

This circuit drives a wind chime using solenoids. It does so in a way that neither affects the tonality of the result, nor prevents the chimes from being operated by the wind in the normal way. So you get the best of both worlds.

More good news is that electronically, it is fairly simple and uses readily-available parts. So you should not have difficulty building it, nor is it likely to break the bank. However, you will need a degree of mechanical skill to make it. Read on to understand why this is so.

Wind chime basics

Wind chimes play a series of notes that are generated by a clapper striking the sides of chime tubes. These tubes hang freely, so they can resonate at their tuned frequency when struck. The clapper is moved by a sail, which is driven by the wind. Fig.1 shows the basic arrangement.

The notes and sounds are very dependent on chime tube length, thickness and diameter and the hanging point.

The frequency is higher with smaller wind chimes – these tinkle away with a light breeze, producing high-pitched notes at a fast rate. Larger wind chimes produce lower-frequency tones at a slower rate.

Wind chime sound quality is also dependent upon the clapper. Its mass, density, shape and what it is made from very much determines what sound you get.

Tonal differences can be demonstrated by tapping the chime tube with various implements such as a screwdriver blade, screwdriver handle and various pieces of timber. Compare the resulting sounds against that produced by the original clapper.

When using a good-quality wind chime, the clapper will enhance the sound. A low-quality wind chime will have the sound spoiled by the clapper, where it produces an entirely different tone to the resonance sound of the chime tube.

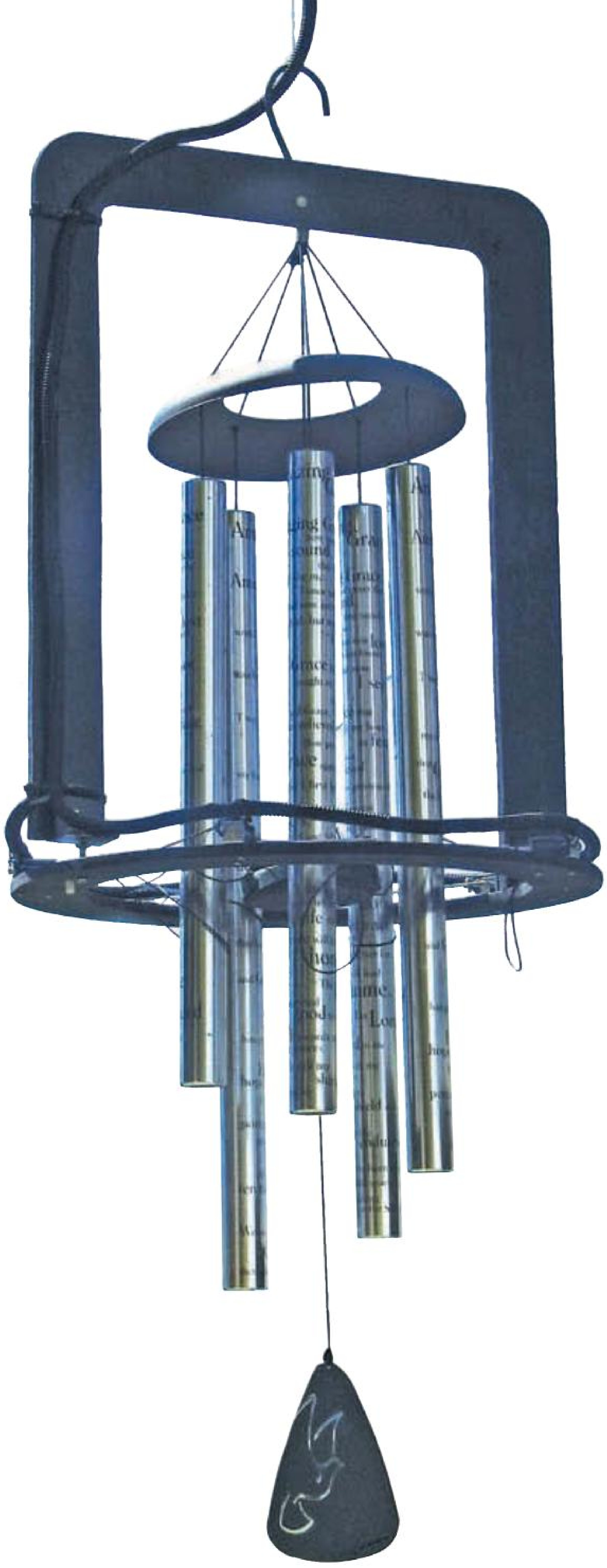
Often, the clapper is a circular piece of timber with a bevelled edge, so that a small area of its side strikes the tube. Timber clappers are much better than metal types. Once struck by the clapper, a chime tube will move away from its resting position due to kinetic energy transfer. The chime tube will resonate to produce sustained tones that differ from the initial strike sound.

If you are after more detail on wind chimes, the science behind them and how to build them, a good site to visit is: **www.leehite.org/Chimes.htm**

This includes calculators to design a wind chime to produce the desired notes. Be aware that the notes perceived from a wind

Features and specifications

- Drives wind chimes with up to 12 elements (or multiple smaller chimes)
- Suits a wide range of sizes from miniature chimes up to large ones
- Individual calibration of solenoid drive control parameters
- Sequence recording and playback
- Sequences with long delays can be recorded in shorter periods
- Optional randomisation of the time between chime strikes
- Adjustable randomisation parameters
- Optional automatic switch-off in darkness



chime can be very different from the fundamental resonance of each chime tube.

Solenoid drive

The biggest challenge in making a solenoid-driven wind chime is in maintaining the original sound quality. While a wind chime could be played using solenoids that directly strike the chime tubes, the sound produced in this manner is rather poor.

A very simple solenoid-driven wind chime arrangement is shown in Fig.2. The solenoid push ends can be arranged to strike the chimes in a straight-line wind chime, which can be made from a disassembled wind chime. While this is easy to build, apart from the poor sound quality, it also has the disadvantage that it can no longer be played by the wind.

A more complex solenoid-driven wind chime, which retains the original configuration, is shown in Fig.3. Good sound quality is maintained by using the solenoids to pull the clapper that, in turn, strikes the tubes in a similar way to when driven by the wind. Additionally, the wind chime is not significantly prevented from its normal operation of playing sounds due to wind.

So building this device involves some electronic assembly, mechanical fabrication and a little woodworking. The electronic side involves the assembly of a circuit board, initial solenoid calibration and other adjustments. On the mechanical side, you need to arrange the solenoids and other bits and pieces to activate the clapper.

The woodworking aspect involves making a frame to support these solenoid movements, which are arranged around the outside of the wind chime.

Design features

Our *Electronic Wind Chime* circuitry can drive up to 12 solenoids, so it can be used to play up to 12 different chimes. These chimes don't have to be within the same wind chime. You could use the same circuitry to control two or more wind chimes, so long as there are no more than 12 chimes in total.

You can also mix and match solenoids – for example, using smaller solenoids for small chimes and larger solenoids for larger chimes. Each solenoid can be independently set up for how it is driven.

There are two adjustments. One controls the voltage applied to each solenoid. This can be varied from the full 12V down to near 0V via pulse width modulation.

This feature is used to prevent the solenoid from being too aggressive. A lower voltage will slow down the solenoid action, so that the wind chime is not sent into disarray.

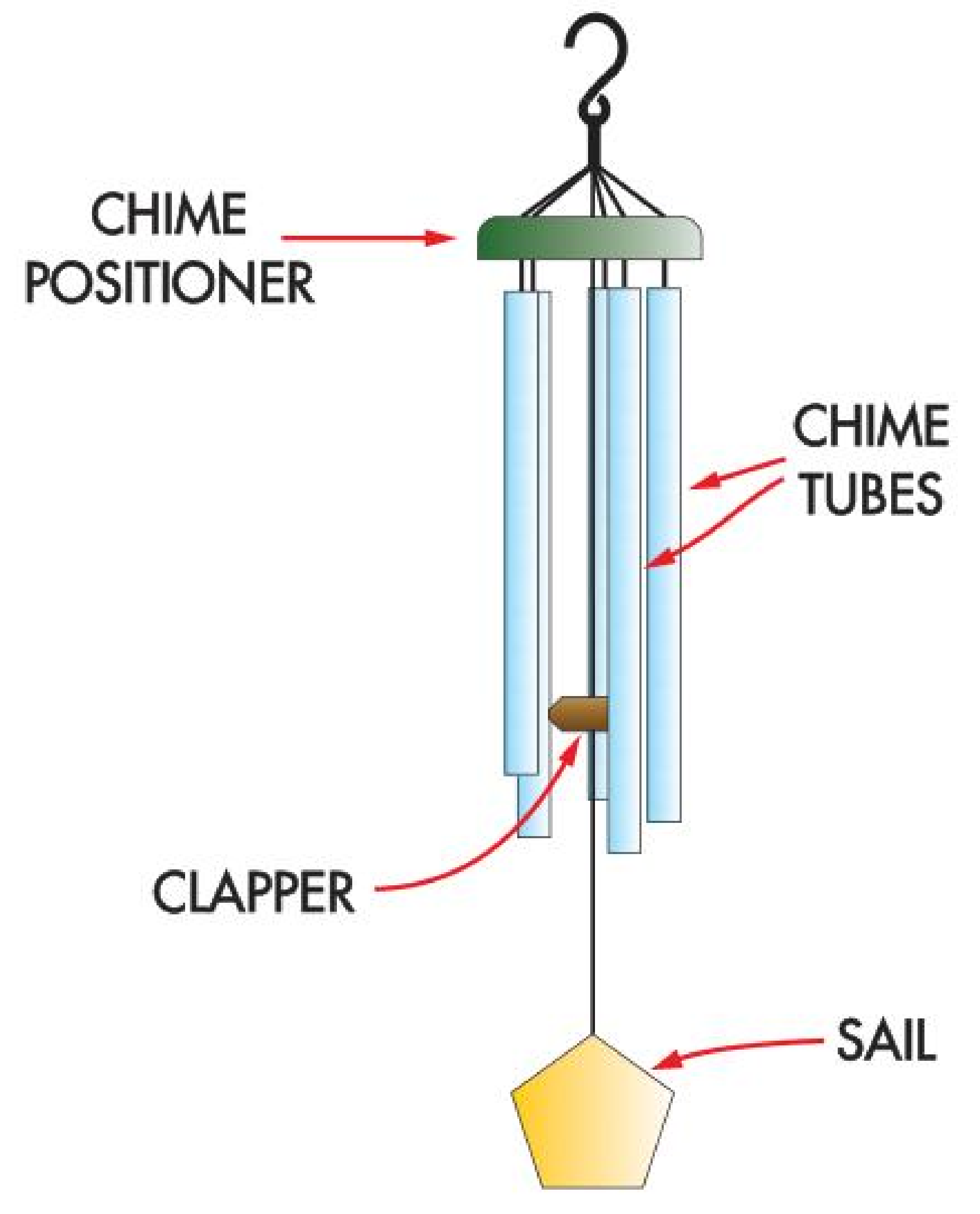


Fig.1: in a standard wind chime, the wind blows the sail which moves the clapper, bringing it into contact with the chime tubes. Each time it strikes a tube, it makes a sound and then bounces off, possibly hitting other tubes. The result is a non-repetitive series of tones, varying with the strength and direction of the wind.

The second adjustment is the duration the solenoid is driven. This needs to be sufficient to allow it to produce a strike against the chime and then pull away before the chime tube returns.

The electronics includes the option to manually 'play' the wind chime by pressing small pushbutton switches. These are useful during calibration, to check whether each chime is being struck correctly. But these switches have another purpose – you can record a sequence by manually playing the solenoids using these buttons, then play it back later, to play a tune (for example).

The sequence of solenoids and the period between each activation is recorded. There is also a facility to record long breaks between solenoid strikes without having to wait the full period. This feature increases the period that's recorded by a factor of 10, so you can record a very long, slow sequence in a reasonable amount of time.

During recording, a variety of different sequences can be included. This will decrease the perceived repetition as the played-back sequence repeats in a loop.

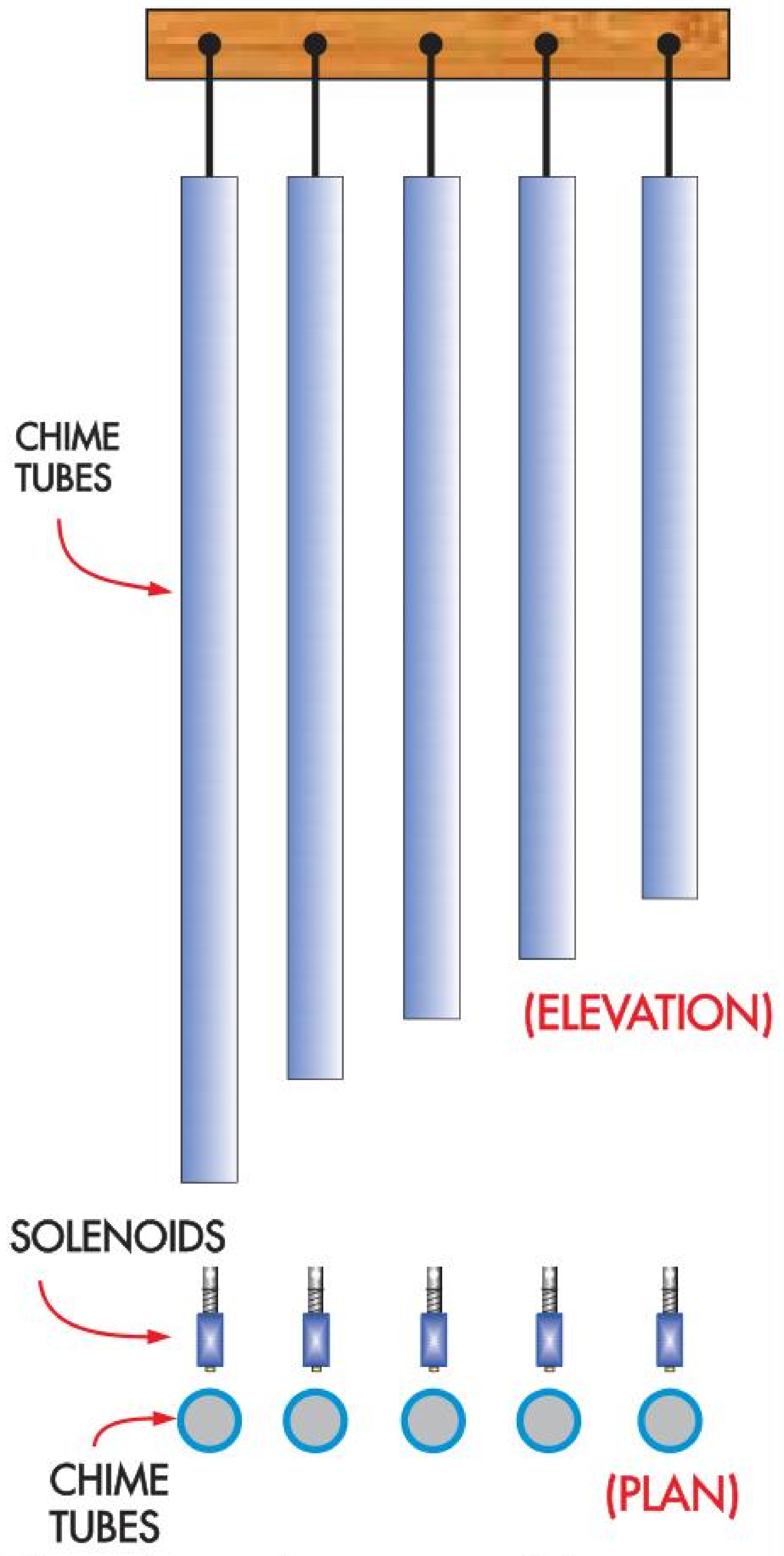


Fig.2: the easiest way to drive a wind chime with solenoids would be to rearrange the tubes in a row and then place a row of solenoids alongside. This is not a very good approach, though, as the solenoid plungers will make a different sound when striking the tubes compared to the (usually timber) clapper. Also, this modified chime would no longer work the same (or possibly at all) when driven by the wind.

The recording time available is well over what you might require. This means that you are free to record without concern of running out of memory. The recording is permanently stored, unless overwritten with a new recording.

There is also an option to randomise the pauses between solenoid strikes during playback. At the maximum randomness setting, the delays vary between one and five times longer than those recorded.

The randomness changes to a new value at intervals of between 10 seconds and 21.25 minutes; this, in itself, varies randomly.

This is all designed to remove any hint of a machine-driven wind chime, making it sound more natural.

The maximum randomness values can be changed to smaller values if desired. Optionally, the *Electronic Wind Chime* can be set to switch off during darkness. This is useful if you (or your neighbors!) prefer peaceful serenity at night.

Circuit details

The circuitry, shown in Fig.4, is based around microcontroller IC1. It stores the recorded sequences in its Flash memory, then plays them back by using its digital outputs to drive transistors or MOSFETs that, in turn, drive the solenoids. The microcontroller also monitors a light-dependent resistor (LDR1), a control switch, jumper link and a trimpot and drives a status LED (LED1).

12 of IC1's 20 pins are used as digital outputs for driving the solenoids. There are two types of solenoid drivers you can use. One option is NPN transistors for driving low-current solenoids.

This is a considerable cost saving compared to N-channel MOSFETs, but MOSFET drivers must be used for solenoids that draw over 500mA.

There is a small circuit change when using a transistor rather than a MOSFET: the resistor value (R1-R12). When a transistor is used, the resistor value is 2.2kΩ, which sets the transistor base current. For a MOSFET, the resistor value is 100Ω instead, and this drives the MOSFET gate.

Diodes D1-D12 at the transistor collector or MOSFET drain are there to conduct the reverse voltage (back-EMF) from the solenoid coil when it is switched off. This protects the bipolar transistor or MOSFET from damage.

PWM drive

The solenoids can be driven with a PWM signal. This is where the MOSFET or transistor is switched on and off at 500Hz with a particular duty cycle. The average voltage produced is the duty cycle multiplied by the supply voltage. So for a 12V supply and a 50% duty cycle, the average voltage applied to the solenoid is 6V.

The frequency needs to be high enough to prevent the solenoid from driving the plunger in and out at the PWM rate. But too high a frequency can also cause problems such as increased dissipation in the transistor/MOSFET or reduced response from the magnetic properties of the steel core.

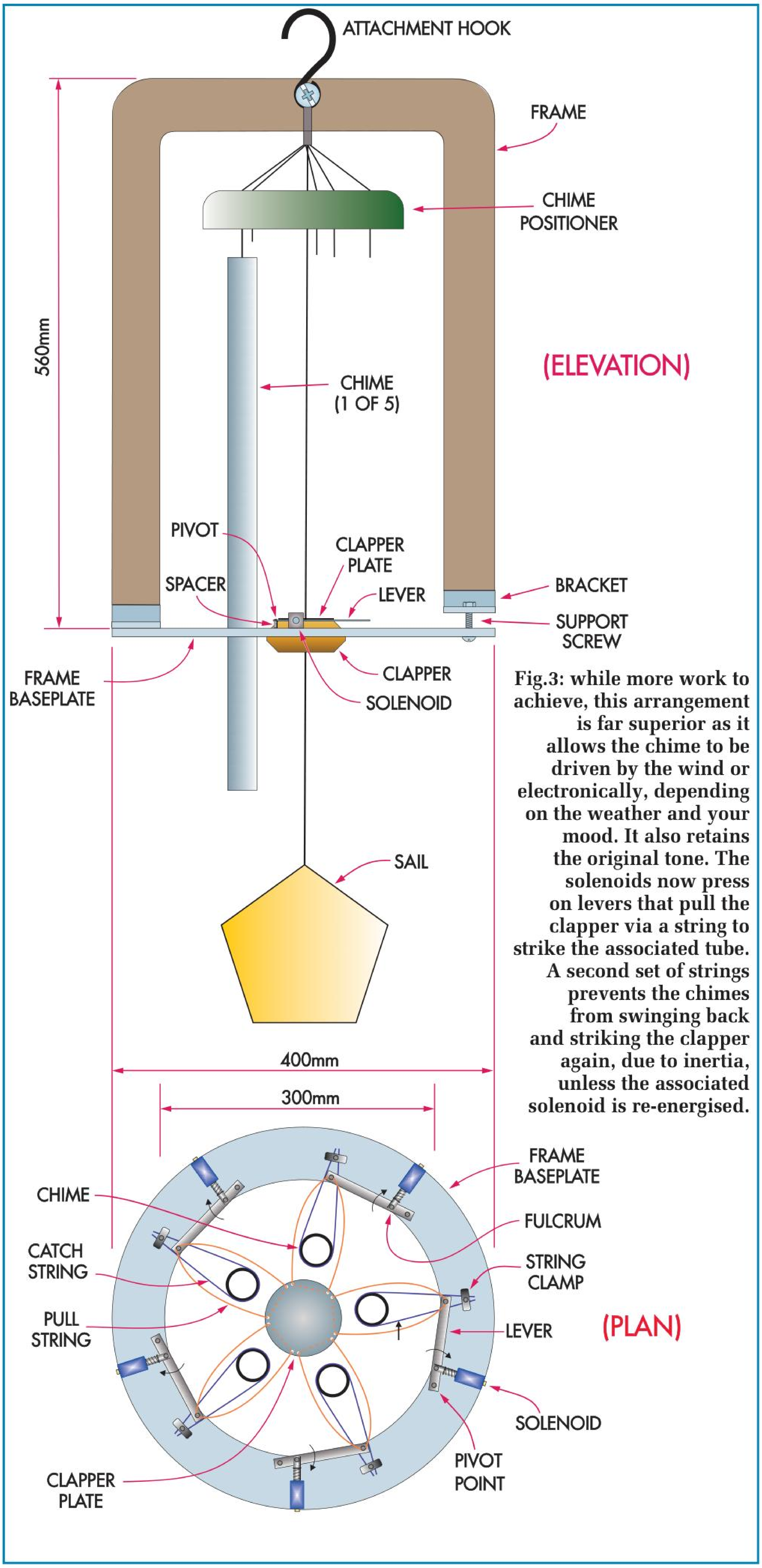
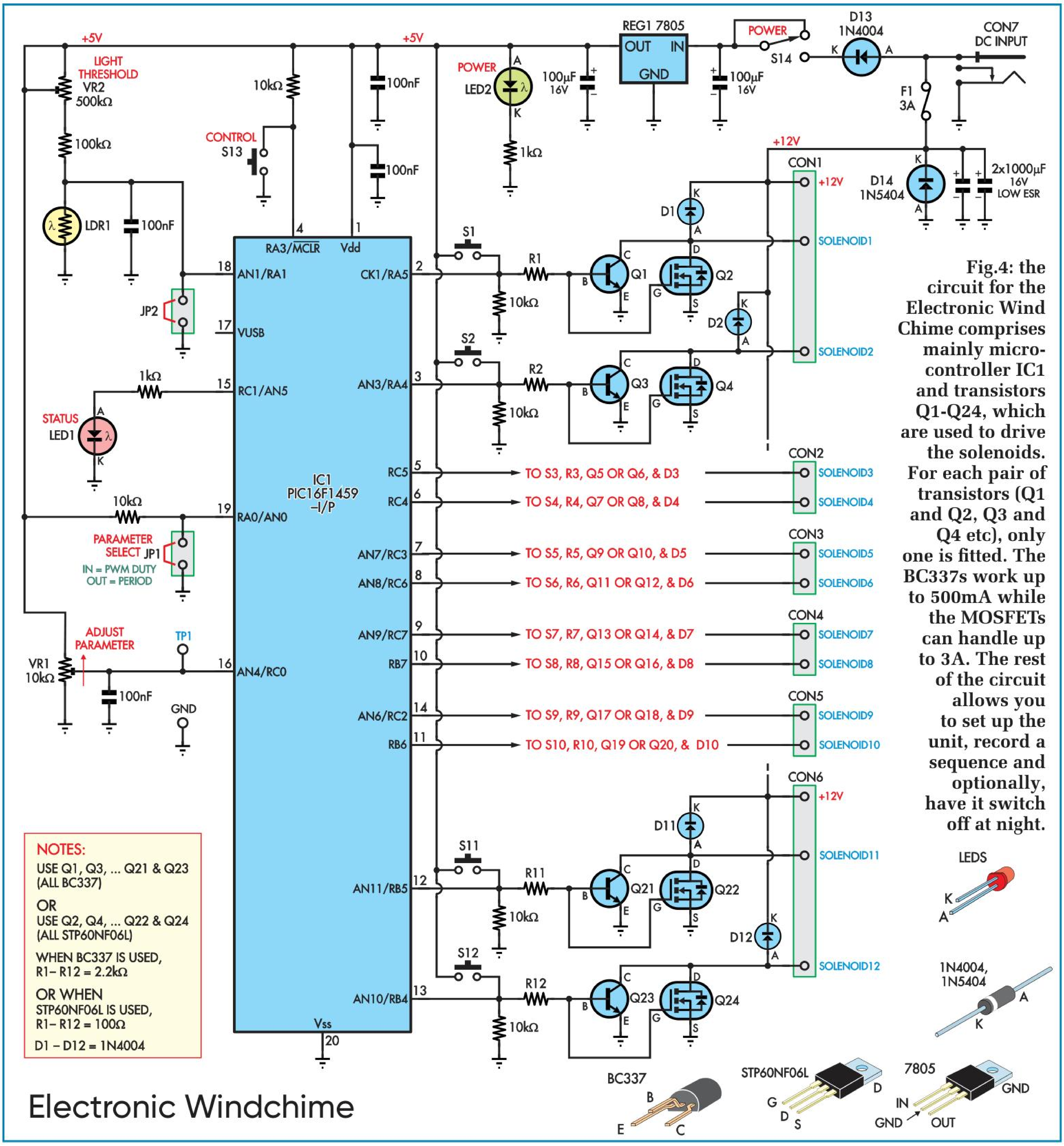


Fig.3: while more work to achieve, this arrangement is far superior as it allows the chime to be driven by the wind or electronically, depending on the weather and your mood. It also retains the original tone. The solenoids now press on levers that pull the clapper via a string to strike the associated tube. A second set of strings prevents the chimes from swinging back and striking the clapper again, due to inertia, unless the associated solenoid is re-energised.


Electronic Windchime
+5V
LIGHT THRESHOLD
VR2 500kΩ
100kΩ
LDR1
100nF
JP2
10kΩ
CONTROL S13
100nF
POWER LED2
100µF 16V
REG1 7805
OUT IN GND
POWER S14
D13 1N4004
CON7 DC INPUT
F1 3A
+12V
CON1
D14 1N5404
2x1000µF 16V LOW ESR
1kΩ
RA3/MCLR
Vdd
AN1/RA1
VUSB
RC1/AN5
RA0/AN0
AN4/RC0
CK1/RA5
AN3/RA4
RC5
RC4
AN7/RC3
AN8/RC6
AN9/RC7
RB7
AN6/RC2
RB6
AN11/RB5
AN10/RB4
Vss
IC1 PIC16F1459 –I/P
STATUS LED1
PARAMETER SELECT JP1
IN = PWM DUTY
OUT = PERIOD
ADJUST PARAMETER
VR1 10kΩ
TP1
GND
S1
R1
Q1
Q2
D1
D2
S2
R2
Q3
Q4
SOLENOID1
SOLENOID2
TO S3, R3, Q5 OR Q6, & D3
TO S4, R4, Q7 OR Q8, & D4
TO S5, R5, Q9 OR Q10, & D5
TO S6, R6, Q11 OR Q12, & D6
TO S7, R7, Q13 OR Q14, & D7
TO S8, R8, Q15 OR Q16, & D8
TO S9, R9, Q17 OR Q18, & D9
TO S10, R10, Q19 OR Q20, & D10
CON2
SOLENOID3
SOLENOID4
CON3
SOLENOID5
SOLENOID6
CON4
SOLENOID7
SOLENOID8
CON5
SOLENOID9
SOLENOID10
CON6
D11
D12
S11
R11
Q21
Q22
S12
R12
Q23
Q24
SOLENOID11
SOLENOID12
LEDS
K A
1N4004, 1N5404
BC337
B E C
STP60NF06L
G D S D
7805
IN GND OUT GND

NOTES:

USE Q1, Q3, ... Q21 & Q23 (ALL BC337)

OR

USE Q2, Q4, ... Q22 & Q24 (ALL STP60NF06L)

WHEN BC337 IS USED, R1– R12 = 2.2kΩ

OR WHEN STP60NF06L IS USED, R1– R12 = 100Ω

D1 – D12 = 1N4004

Fig.4: the circuit for the Electronic Wind Chime comprises mainly microcontroller IC1 and transistors Q1-Q24, which are used to drive the solenoids. For each pair of transistors (Q1 and Q2, Q3 and Q4 etc), only one is fitted. The BC337s work up to 500mA while the MOSFETs can handle up to 3A. The rest of the circuit allows you to set up the unit, record a sequence and optionally, have it switch off at night.

Our choice of 500Hz was suitable for a wide variety of solenoids that we tested.

Oscilloscope waveform Scope1 shows the gate drive to the MOSFET at the top (yellow) with a 5V drive voltage. The drain voltage waveform (blue) is the lower trace with a 12V supply voltage. The solenoid has 12V across it when the drain voltage is 0V, and 0V across it when the drain is at 12V (the negative end of the solenoid connects to the drain). The duty cycle is around 50% at almost 500Hz.

The solenoid driver pins on IC1 usually are set as inputs. The MOSFET or transistor is held off via the associated 10kΩ pull-down resistor. Having the pins as inputs allows switches S1-S12 to pull the input high when pressed. If the pin were set as a low output instead, the pull-up switch would 'fight' the microcontroller output, causing a high current through the output pin.

The pin is changed to a high-level output when required to switch on the MOSFET or bipolar transistor. In this case, pressing the associated switch will not cause problems since the output is already high.

For a low level, the pin is made an input again, so the MOSFET or bipolar transistor switches off (unless the associated switch is currently being pressed).

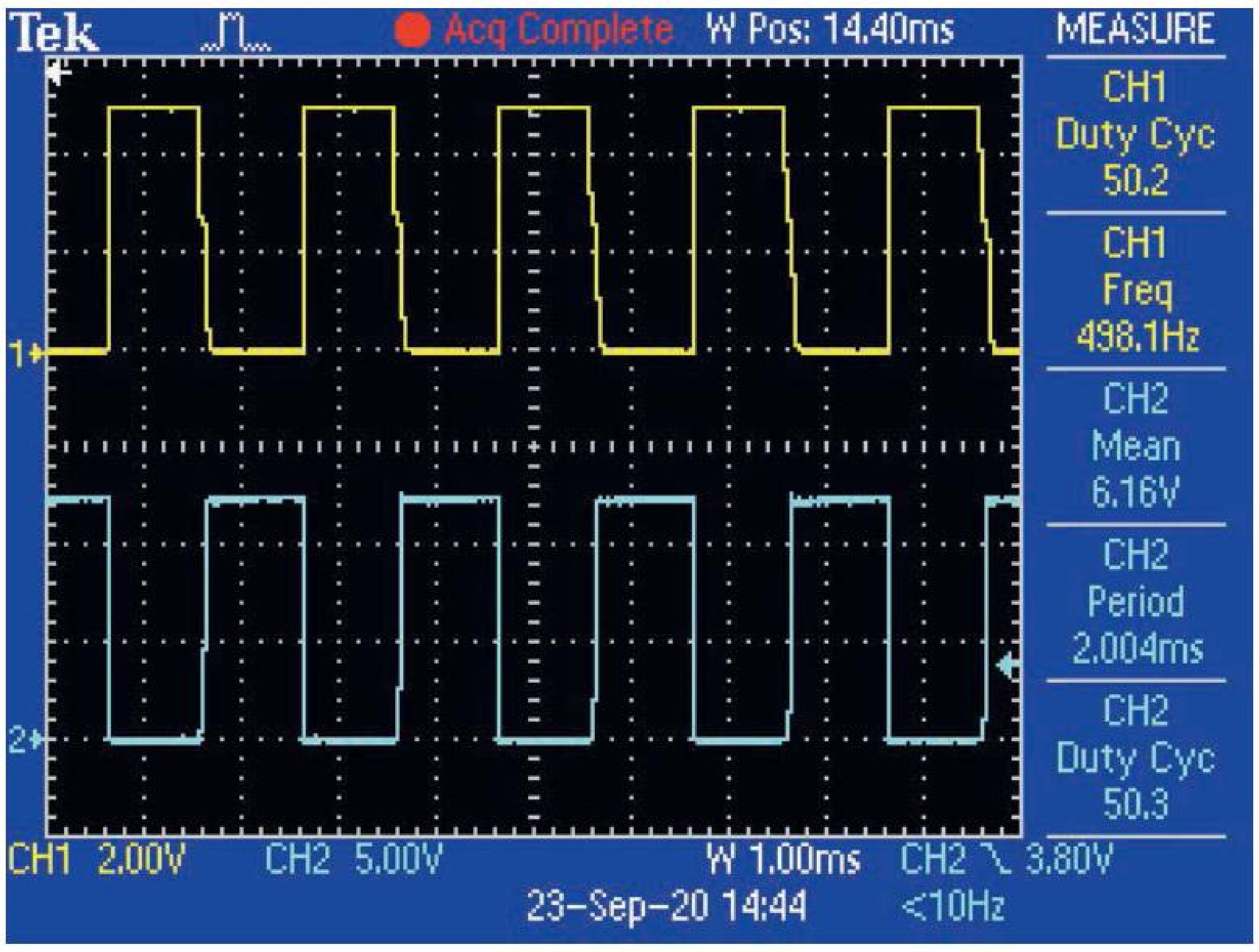
Note that pressing switches S1-S12 causes the associated solenoid to be driven with the full 12V for the duration that the switch is pressed. But when the solenoid is driven via the microcontroller, the drive is a PWM waveform with a preset on-period and duty cycle.

More circuit details

IC1's pin 18 (digital input RA1) monitors the LDR so that the circuit can optionally switch off at night. During the daytime, the LDR resistance is low, so pin 18's voltage is below the low threshold of the RA1 input. A 100kΩ resistor and trimpot VR2 form a voltage divider with the LDR across the 5V supply.

This trimpot allows the detected light threshold to be varied. When the LDR is in darkness, the LDR resistance is high, and this pull-up resistance causes the

Scope1: the 500Hz, 5V PWM drive to the base/gate of the output transistor is shown in the top trace (yellow) with a 50% duty cycle, and the resulting (inverted) 12V drive voltage to the solenoid is shown below in cyan. The duty cycle (ie, percentage of time that the solenoid receives current) is adjustable for each solenoid, to control how hard it is driven.



RA1 voltage to be above its high threshold. IC1 detects this, and the software stops running.

The RA3 digital input monitors control switch S13. This pin can be used as an external master clear signal ($\overline{\text{MCLR}}$) or a general-purpose input. We are using it as an input, and it is usually pulled high, to 5V, by the 10kΩ resistor. This input goes low when the switch is pressed; it serves many functions, as described later.

The status LED (LED1) is driven via the RC1 output via a 1kΩ resistor. It is used to indicate various modes when recording a sequence and calibrating the solenoid settings.

Trimpot VR1 is connected across the 5V supply, and its 0-5V wiper voltage is monitored at IC1's analogue input AN4 (pin 16). VR1 sets the solenoid pulse width/duty cycle and drive duration in conjunction with jumper JP1. JP1 is monitored by IC1's RA0 digital input (pin 19).

This input is held high by the 10kΩ pull-up resistor unless there is a shorting link across JP1, which would pull it low.

Power supply

12V power for the circuit is applied at CON7. This flows to the solenoids via fuse F1. This supply is bypassed with two parallel 1000µF low-ESR capacitors, which helps to supply the peak solenoid current.

Reverse polarity protection uses a 3A diode, D14. If the supply is connected backwards, this conducts to blow the fuse.

The voltage to the remainder of the circuit is applied via reverse polarity protection diode D13, and is switched by S14 before being applied to the input of the 5V regulator, REG1. Two 100µF capacitors, one at the regulator input and the other at the output improve the regulator's stability and transient response. Microcontroller IC1 also has two 100nF supply bypass capacitors at pins 1 and 20.

LED2 lights up when power is applied, with its current limited to around 2-3mA by its 1kΩ series resistor.

Memory storage

Twelve bytes of the Flash memory are dedicated to storing the PWM duty cycle and on-period parameters for each solenoid (ie, one byte per solenoid).

1182 bytes of Flash memory are used for storing the playback sequence. Two bytes of memory are used to record which solenoid(s) to activate, followed by a two-byte delay period. Each delay period can be up to 10.9 minutes in 10ms steps. If the delay period is over 10.9 minutes, then the next two bytes continue that delay.

This means that the maximum sequence can be up to 107 hours (1182 ÷ 2 × 10.9 minutes). However, as extra bytes are consumed for each solenoid strike, the practical maximum is somewhat less than that.

For a more realistic calculation, say that a recording consists of a series of eight strikes, spaced two seconds apart, with a 10-second delay before the next little tune.

That consumes 32 bytes (8 × 4 bytes) for every 24 seconds of recording (7 × 2 seconds + 10 seconds). The 1182 byte memory can record up to 37 such sequences, for a total recording or playback time of 888 seconds (or 14.8 minutes).

Typically, you would leave a longer period between solenoid drive sequences, so the maximum recording (and hence playback) time will be longer. There is no need to completely fill the memory, as during playback, it only cycles through the number of bytes that were recorded in memory

PCB assembly

The *Electronic Wind Chime* circuit is built on a PCB coded 23011201 which measures 147 x 87.5mm and is available from the *PE PCB Service* – see Fig.5. This fits into a UB1 Jiffy box. Which parts you install depends to some extent on the number of solenoids you will use and the solenoid sizes. See the accompanying panel on this topic.

The parts list specifies the parts required to drive the maximum 12 solenoids. Asterisks indicate which parts you can buy fewer of if you plan to drive a smaller number of solenoids. This includes S1-S12, R1-R12, the 10kΩ pull-down resistors, Q1-Q24, D1-D12 and CON1-CON6.

CON1 and CON6 are three-way terminal blocks, with two terminals for a pair of solenoids plus a common positive connection for each set of six. CON2-CON5 are two-way terminal blocks which do not have the common positive connection, only the negative connections for two solenoids. So if you have an odd number of solenoids, you will end up with an unused terminal in one of the connectors.

You can have a mix of low- and high-current solenoid drivers. Say you might wish to control two wind chimes, with each having three large chimes and three smaller ones.

You could fit MOSFETs at the even-numbered positions (Q4, Q8, Q12...) and corresponding 100Ω gate resistors. You'd then fit transistors at the odd-numbered

This PCB has five high power MOSFETs in positions Q2-Q10 with seven lower-power transistors in Q11-Q23. The reason (and difference) is explained in the text. The PCB mounts in the case without screws – it simply clips into the slots on the side guides. As yet, the holes are not drilled into the lid for the on/off switch nor LED – these can be done using the front panel artwork as a template. We'll look at this in more detail next month.

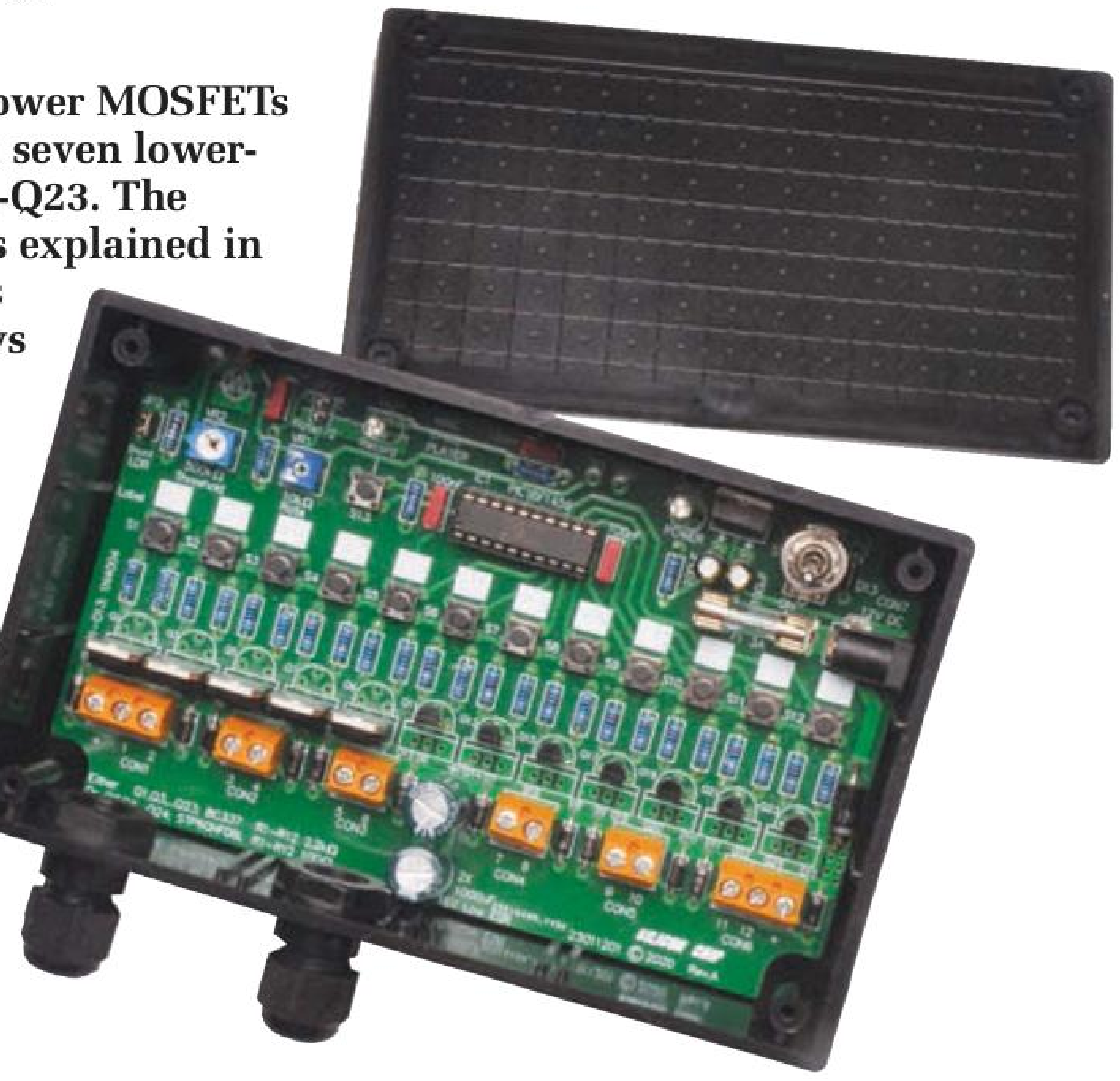
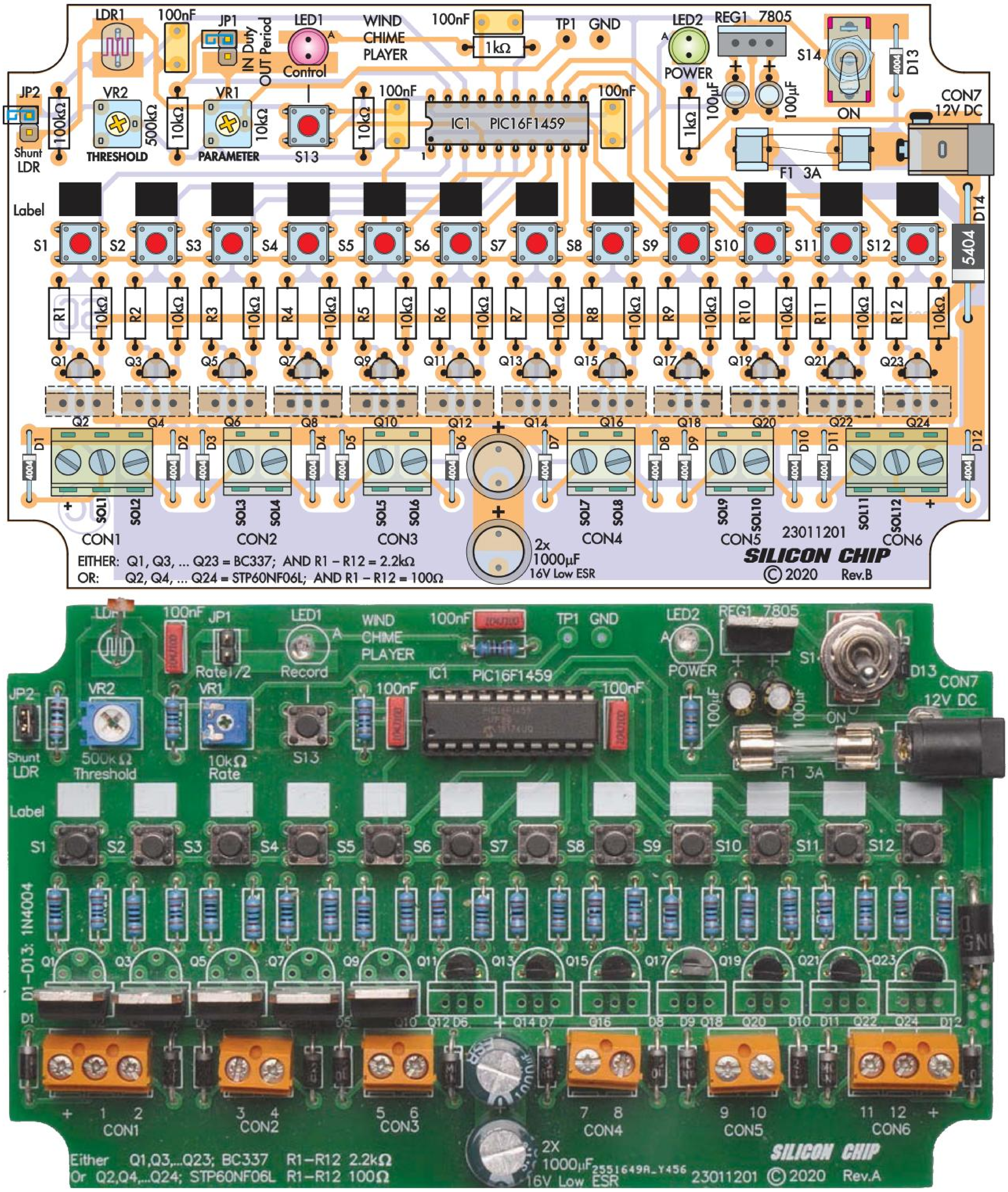


Fig.5: circuit board assembly is straightforward; simply install the components as shown here. Small rectangles are provided above the manual control switches so you can write the musical note produced by that switch, or a solenoid number. During construction, take care with the orientations of the diodes, ICs, transistors, terminal blocks and electrolytic capacitors.


LDR1
100nF
JP1
IN Duty
OUT Period
LED1
Control
WIND CHIME PLAYER
100nF
1kΩ
TP1
GND
LED2
POWER
REG1 7805
S14
ON
D13
CON7 12V DC
JP2
Shunt LDR
100kΩ
VR2
500kΩ
THRESHOLD
10kΩ
VR1
10kΩ
PARAMETER
S13
100nF
IC1 PIC16F1459
100nF
1kΩ
100µF
100µF
F1 3A
Label
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
5404
D14
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
10kΩ
Q1
Q3
Q5
Q7
Q9
Q11
Q13
Q15
Q17
Q19
Q21
Q23
Q2
Q4
Q6
Q8
Q10
Q12
Q14
Q16
Q18
Q20
Q22
Q24
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
4004
SOL1
SOL2
SOL3
SOL4
SOL5
SOL6
SOL7
SOL8
SOL9
SOL10
SOL11
SOL12
CON1
CON2
CON3
CON4
CON5
CON6
2x 1000µF 16V Low ESR
EITHER: Q1, Q3, ... Q23 = BC337; AND R1 – R12 = 2.2kΩ
OR: Q2, Q4, ... Q24 = STP60NF06L; AND R1 – R12 = 100Ω
23011201
SILICON CHIP
© 2020 Rev.B
Rate 1/2
Record
500kΩ Threshold
10kΩ Rate
D1–D13: 1N4004
Either Q1,Q3,...Q23; BC337 R1–R12 2.2kΩ
Or Q2,Q4,...Q24; STP60NF06L R1–R12 100Ω
2X 1000µF 16V Low ESR
23011201 © 2020 Rev.A

Q position (Q1, Q5, Q9...) with 2.2kΩ base resistors, for the smaller chimes. Do not install both a MOSFET and bipolar transistor in the same position.

This complicates construction a little, but you can save quite a bit of money as the bipolar transistors cost far less than the MOSFETs.

Start by fitting the resistors on the PCB where shown (remember to vary the R1-R12 as described above). It's always best to check the values with a digital multimeter (DMM) set to measure resistance.

Continuing on, install diodes D1 to D12 (or as many as required) and D13. Make sure that the cathode stripes face toward the top of the PCB as shown. Also fit D14 now, which faces the opposite direction compared to the others, and is the largest diode.

Then mount switches S1-S12 (where used) and S13. These will only fit onto the PCB the right way, so if the switch does not seem to fit, try rotating it by 90°.

We recommend that IC1 is installed using a socket. Make sure the end notch faces toward the left edge of the PCB.

The trimpots can be installed next. VR1 is the 10kΩ trimpot that may be marked as '103' rather than '10k'. VR2 is 500kΩ and may be marked as '504' rather than '500k'.

Now mount the fuse clips, making make sure these are installed with the correct orientation, ie, with the end stops toward the outside of the fuse. It is a good idea to insert the fuse before soldering the clips to ensure the fuse is aligned within the clips, and that the clips are oriented correctly.

PC stakes can also be installed at GND and TP1. However, these can be left out, and multimeter probes pressed directly onto the pads for voltage measurements.

Fit the two-way headers for JP1 and JP2 next, then the DC socket (CON7). Follow with the 3-way and 2-way screw terminals (as many as needed), with the wire entry holes towards the lower edge of the PCB.

Now mount the capacitors, noting that the electrolytic capacitors must be oriented correctly, with the longer positive leads through the holes marked '+'.

Transistors

It is now time to fit the transistors and/or MOSFETs (along with regulator REG1), noting again that which ones and how many you install depends on what solenoids you are using, and how many.

The power switch (S14) and the two LEDs can be mounted in one of two ways: either directly on the PCB or onto the lid of the box, with wires making the connections between the component and PCB.

We opted to mount the switch and LEDs on the PCB – this way, they will not be seen or accessible once the lid of the box is in place, but that's OK as they are mainly used during setup and recording. Without the power switch being accessible, the unit can still be switched on and off via the 12V plugpack.

If you intend to use the LDR to switch the unit off at night, solder this in place now. It can be mounted so that the face of the LDR is toward the back

Parts List – Electronic Wind Chimes

1 double-sided plated-through PCB coded 23011201, 147 x 87.5mm
[available from the PE PCB Service]
1 UB1 Jiffy box, 158 x 95 x 53mm
[Jaycar HB6011 (black), Altronics H0201 (black) or H0151 (grey)]
1 12V DC plugpack or similar supply, ideally with 2.5mm ID barrel plug
(current rating dependent on solenoids used, up to 3A maximum)
12* 12V DC spring-return pull solenoids with lever slot [see text]
2* 3-way screw terminals with 5.08mm spacing (CON1,CON6)
4* 2-way screw terminals with 5.08mm spacing (CON2-CON5)
12* SPST momentary switches (S1-S12) [Altronics S1120, Jaycar SP0600]
1 SPST momentary switch (S13) [Altronics S1120, Jaycar SP0600]
1 SPDT toggle switch (S14) [Jaycar ST0335, Altronics S1310]
2 M205 PCB-mount fuse clips (F1)
1 3A M205 fast blow fuse (F1)
1 5A DC PCB-mount 2.5mm ID barrel socket (CON7)
[Jaycar PS0520, Altronics P0621A]
1 20-pin DIL IC socket (for IC1)
1 48kΩ to 140kΩ light-dependent resistor (LDR1)
[Jaycar RD3480, Altronics Z1619]
2 2-way pin headers with jumper shunts (JP1,JP2)
2 PC stakes (optional; GND and TP1)
2 or more cable glands for 3-6.5mm cable entry

Semiconductors
1 PIC16F1459-I/P 8-bit microcontroller programmed with 2301120A.hex (IC1)
1 7805 1A 5V regulator (REG1)
1 3mm red LED (LED1)
1 3mm green LED (LED2)
12* 1N4004 1A diodes (D1-D12)
1 1N4004 1A diode (D13)
1 1N5404 3A diode (D14)

** reduce these quantities for driving fewer than 12 solenoids and note that low- and high-current solenoid drivers can be mixed and matched (up to a total of 12)*

Capacitors
2 1000µF 16V PC low-ESR electrolytic
2 100µF 16V electrolytic
4 100nF MKT polyester

Resistors (all 1/4W 1% metal film
1 100kΩ (Code brown black black orange brown)
12* 10kΩ (S1-S12 pull-down resistors) (Code brown black black red brown)
2 10kΩ (Code brown black black red brown)
2 1kΩ (Code brown black black brown brown)
1 500kΩ miniature horizontal trim pot, Bourns 3386P style (VR2) (Code 504)
1 10kΩ miniature horizontal trim pot, Bourns 3386P style (VR1) (Code 103)

Parts for high current solenoid drivers (80mA)
12* STP16NF06L, STP60NF06L or CSD18534KCS 60V, 16/60/73A logic-level
N-channel MOSFETs (Q2,Q4,Q6...Q24)
[Jaycar ZT2277]
12* 100Ω 1/4W 1% metal film resistors (R1-R12) (Code brown black black black brown)

Parts for low -current solenoid drivers (80mA)
12* BC337 NPN 500mA transistors (Q1,Q3,Q5...Q23)
12* 2.2kΩ 1/4W 1% metal film resistors (R1-R12) (Code red red black brown brown)

Miscellaneous
Suitable exterior board or timber, aluminium sheet, wire loom, cable ties, wire, screws, paint, string

edge of the PCB (by bending the leads), so it is exposed to the outside light via a hole in the side of the enclosure. If you don't need the LDR feature, link it out or place a shorting block over jumper JP2.

Housing

The PCB is held in the plastic case by the integral clips holding the sides of the PCB.

You will need to drill holes in the box for the DC socket and the solenoid wiring. We recommend that this wiring passes through several cable glands before being connected to CON1-CON6.

The 9mm hole for the DC socket is 21mm above the outside base of the case and 26mm in from the outer edge. Cable glands can be placed 15mm down from the top edge of the enclosure, adjacent to the screw connectors CON1-CON6.

Choosing solenoids

The circuit has been designed to cater for many types of solenoids. We used D-frame spring-return pull types, although push-pull types can also be used. The sizes available range from miniature through to heavy-duty types that can draw up to 3A.

What you need depends on the size of the wind chime you are using. There are several specifications you need to look for; for example, the circuit requires 12V solenoids. Another important specification is the movement length, or stroke.

Other useful features are a means to attach to the solenoid plunger. Some will have holes in the plunger, but others will not have any means to attach anything to the solenoid plunger.

For small wind chimes, a solenoid stroke of 4mm might be sufficient, but for larger chimes, something like 12mm is required.

For use with mini wind chimes (tubes around 6.35mm in diameter) and using a direct solenoid plunger hit to an inline set of chimes as shown in Fig.2, a push-pull solenoid with a frame section that measures 21 x 11 x 10mm having a 4mm stroke would be suitable. Their overall length is 30mm, and they draw 120mA at 12V DC.

The solenoids for the wind chime we used have a 30 x 16 x 14mm frame section and 10mm stroke. Their overall length is 55mm. The plunger includes a mounting slot and securing hole suitable for a lever attachment. At 12V DC, they draw 2A. The initial pull is 300g with an ultimate retention force of 3kg when fully closed.

Both Jaycar and Altronics sell suitable solenoids, and many others are available via on-line marketplaces such as eBay.

Next month

The electronics section is virtually complete, but we still need to describe how to modify your wind chime to add the solenoids, plus the testing, setup and sequences recording procedures. All that will all be covered in next month's article.

Reproduced by arrangement with
SILICON CHIP magazine 2022.
www.siliconchip.com.au

PicoLog Cloud®

View live and saved captures
from PicoLog data loggers
anywhere in the world,
on any device, for **FREE!**

Learn more at
www.picolog.com

Visit **www.picotech.com/A1030** to find out more

Email: sales@picotech.com. Errors and omissions excepted.
Please contact Pico Technology for the latest prices before ordering.

pico® Technology

Using Cheap Asian Electronic Modules

By Jim Rowe

Mini Digital AC Panel Meters

239 V
6.41 A
voltage
current
1536 W
119 Wh
power
energy
237.2 V
6.32 A
1502.0 W
50.0 Hz
0.10 kwh
1.00 PF

In this follow-up article on low-cost digital panel meters, we look at meters designed to measure AC voltages and currents. Some of them can calculate and display power, energy consumption and frequency. As usual, we'll give you an idea of how they work, how they perform and how easy they are to use.

As promised in December 2021's issue, this second article describes some small meters designed to measure AC voltages and currents. The AC models are even more interesting than those we described last time. For a start, they vary more significantly in both size and price.

Like the DC meters we looked at in the first article, these AC meters are all designed to be powered from the same source used for voltage measurements. So no separate power source is needed.

It's important to make sure all connections are properly insulated when taking measurements.

As explained in December, DC meters measure currents by measuring the voltage drop across a very low resistance current shunt. In contrast, AC meters typically measure currents by using a special kind of transformer: a 'current transformer' or 'CT'. This steps down the current to a much lower level, as well as providing galvanic isolation for improved safety.

Current transformer basics

One of the big advantages of AC compared with DC, as Nikola Tesla and George Westinghouse stressed over 120 years ago, is that with AC you can use transformers to step the voltage up or down to whatever level best suits your purposes.

This means that AC power can be stepped up to hundreds of thousands of volts to reduce losses when conveyed over long distances, then stepped down to much lower voltages like 230V or 115V, for somewhat safer use in houses, factories and offices.

Of course, when a transformer steps up the voltage, it also steps down the current, and vice versa. This is due to the conservation of energy (ie, the product of voltage and current at the output must be similar to that at the input). So if the voltage is stepped up by a factor N, the current is stepped down by the same factor, and if the voltage is stepped down by N, the current is stepped up by the same factor. (This is much harder to do with DC; generally, it means converting the DC voltage to AC, stepping it up/down, then rectifying and filtering it to turn it back into DC – not easy to do efficiently!)

The current transformer works on the same principle, as shown in Fig.1. It consists of a toroidal magnetic core, usually made from either silicon steel or ferrite, through which passes the wire carrying the current to be measured. The wire effectively forms the transformer's single-turn primary 'winding'.

Many turns of much lighter wire are wound around the toroidal core to form the transformer's secondary winding. So the turns ratio is 1:N, where N is the number of secondary turns.

When a relatively heavy alternating current is flowing through the wire forming the CT's primary, this produces an alternating magnetic field in its core. And as a result, an AC voltage is induced in the CT's secondary winding, which can provide an alternating current N times smaller than that flowing through the single-turn primary (assuming that it's connected

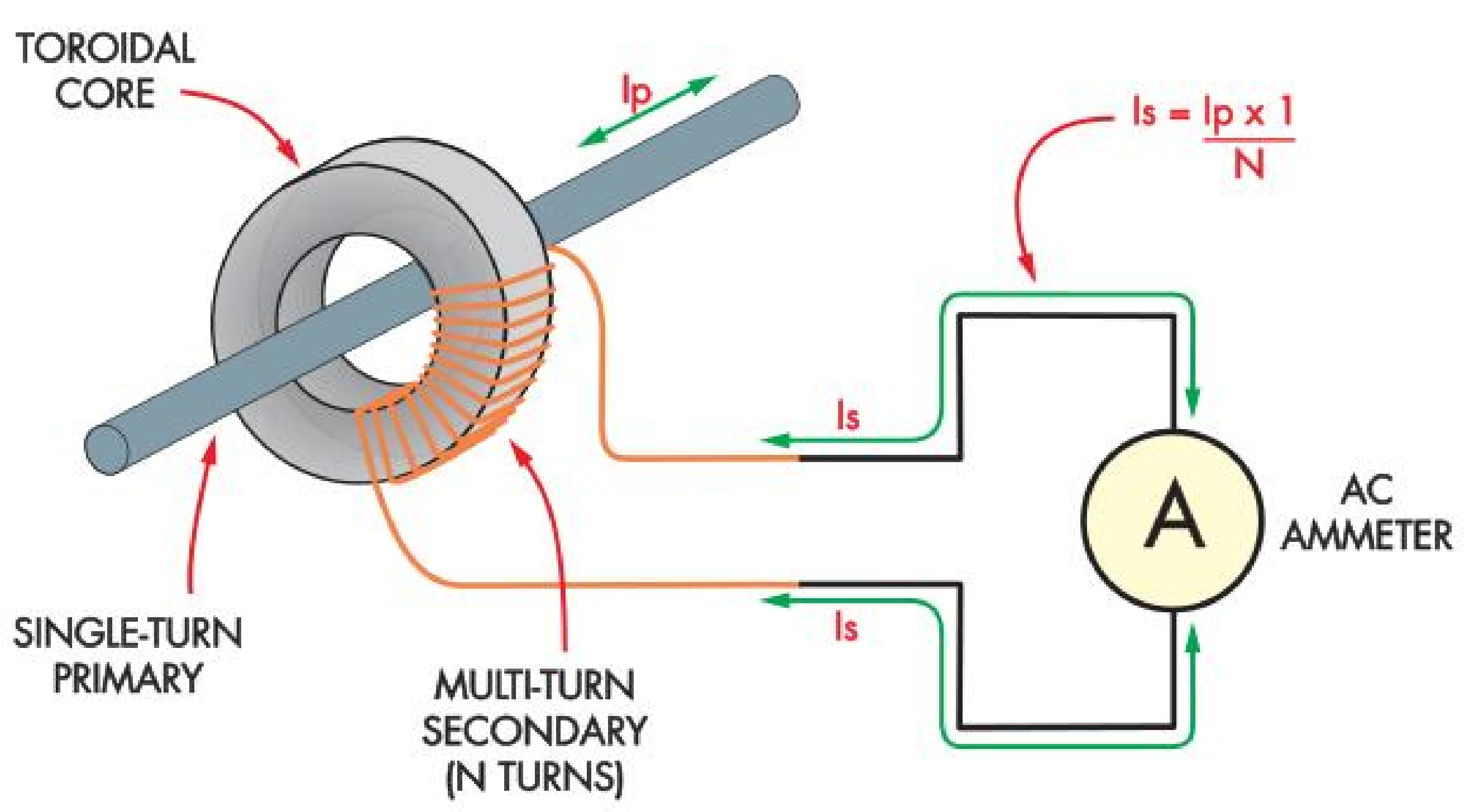
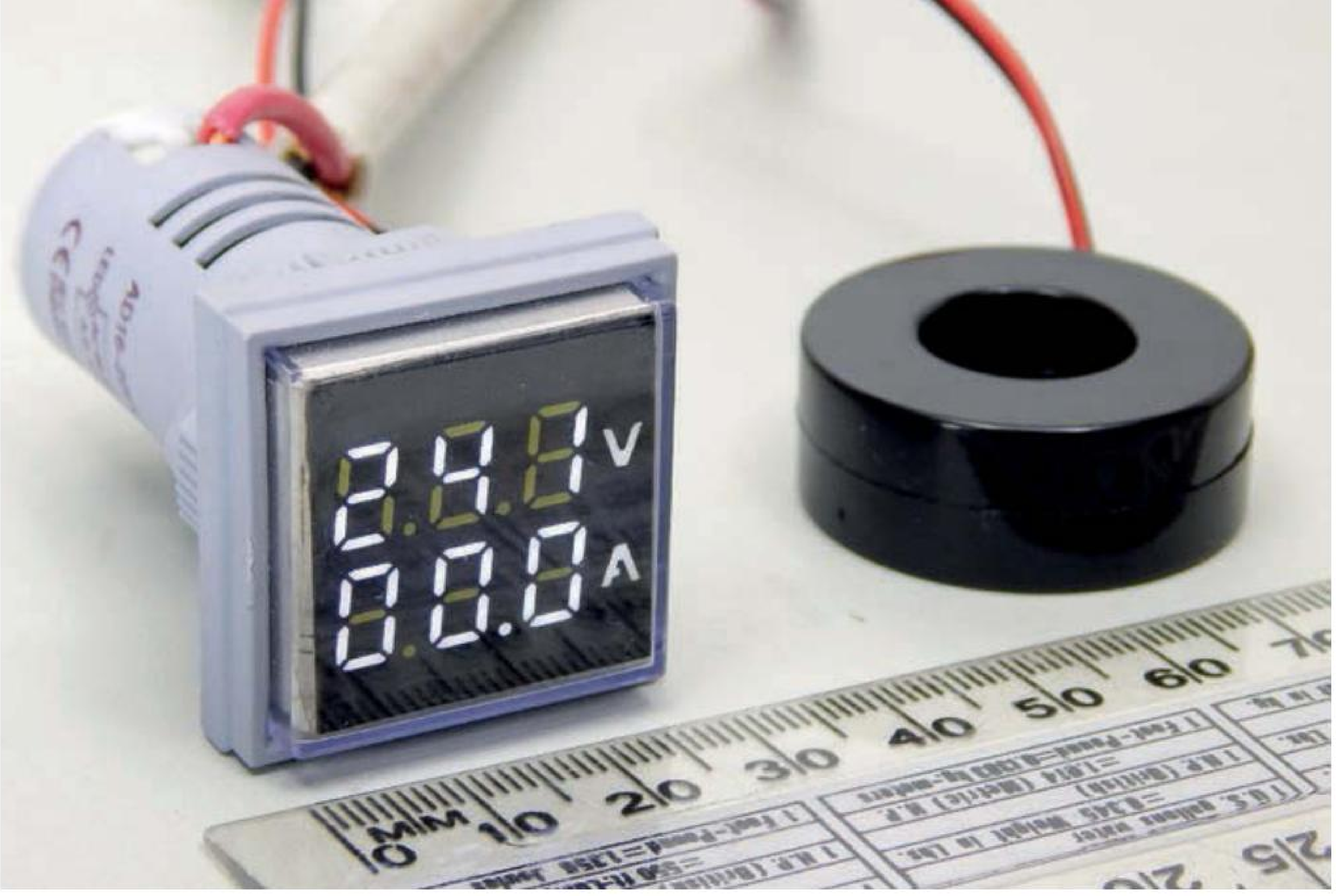

TOROIDAL CORE
I_p
$I_s = I_p \times \frac{1}{N}$
I_s
A
AC AMMETER
I_s
SINGLE-TURN PRIMARY
MULTI-TURN SECONDARY (N TURNS)

Fig.1: how the current transformer (CT) operates. The CT secondary should be terminated with a low impedance, otherwise it will generate a very high voltage if any significant AC current is flowing in the primary. Make sure to connect the secondary leads of the CT to the panel meter before any current is allowed to flow through the primary.



The AD16-22FVA is the smallest AC panel meter out of the three but has the highest measurement range of 60-500V. The current transformer (CT) is shown adjacent and is rated at 0-100A.

The AD16-22FVA meter shown at actual size.

to a low-impedance load or 'burden'). This is illustrated by the expression at upper right in Fig.1, relating secondary current I_S to primary current I_P.

So, for example, if the CT has a secondary winding of 1000 turns and the current flowing in the primary wire is 50A, the secondary current will be 50mA (50A ÷ 1000).

The advantages of using a CT includes a stable transformation ratio, which helps ensure measurement accuracy, as well as a high degree of electrical isolation. The main disadvantage is that the 'primary' wire must be passed through the centre of the transformer core.

One way around this is to have the core in two halves. But this adds significantly to the cost, as well as reducing its conversion efficiency a little (due to the inevitable air gaps).

The AD16-22FVA meter

The AD16-22FVA is both the physically smallest meter that we will describe in this article, and also the lowest in cost.

As you can see from the photos, it's quite tiny, measuring only 31mm wide, 31mm high and 56mm deep. Behind the front square display section, the body is cylindrical so it can pass through a 22mm diameter hole in a panel. It comes with a matching plastic 'nut' that allows the meter to be attached firmly to the panel.

The CT is separate and is connected to the meter via a light two-wire lead. The CT lead is close to 180mm long, while the meter's own lead is 100mm long.

The AD16-22FVA has two 3-digit 7-segment LED displays, one above the other, with both sets of digits 7mm high. And the meter is available in five versions, with red, blue, green, yellow or white displays. It's hard to be sure, but I suspect that all these versions differ only by having different colour filters in front of the same white LED displays. The voltage measurement range of all versions is 60-500V AC, while their current range is 0-100A.

The AD16-22FVA meter is very easy to set up and use, as you can see from Fig.2. All you have to do is pass one of the load power leads through the centre of the CT, and then connect the power terminals of the meter to the same source of AC power.

I found the AD16-22FVA advertised on AliExpress by the supplier Sevenstar Tools at $4.58 plus 78¢ for delivery (including the CT). The 'white display' version I ordered arrived about 30 days later, in good condition.

I checked its performance with my reference instruments, using a finned oil heater as the load. It gave voltage readings that were 0.2% low and current readings that were 0.94% low, compared with my Agilent U1251B DMM. So the AD16-22FVA may be tiny, but its performance is quite respectable.

I admit that I found the small 3-digit displays a little hard to read. But for less than $5.50, it still represents excellent value.

The DL69-2042 meter

Apart from the separate CT, the DL69-2042 AC meter looks almost identical to the DSN-VC288 DC meter we checked out in the last article. It's somewhat larger than the AD16-22FVA at 80mm wide, 42mm high and 48mm deep. It clips into a 75 x 39mm rectangular hole in a panel.

The DL69-2042 sports two 4-digit 7-segment LED displays, both with digits 10mm high. The volts display is at the top, with a red filter, while the current display is below with a green filter.

This meter has a voltage range of 80-300V, although it is also available with a range of 200-450V. In both cases, the current range is 0-100A. The rated accuracy is ±1%, ±2 digits for both voltage and current.

I found the DL69-2042 advertised on the Banggood website for $17.00 plus $3.73 air parcel shipping (again, including the CT), ie, about four times

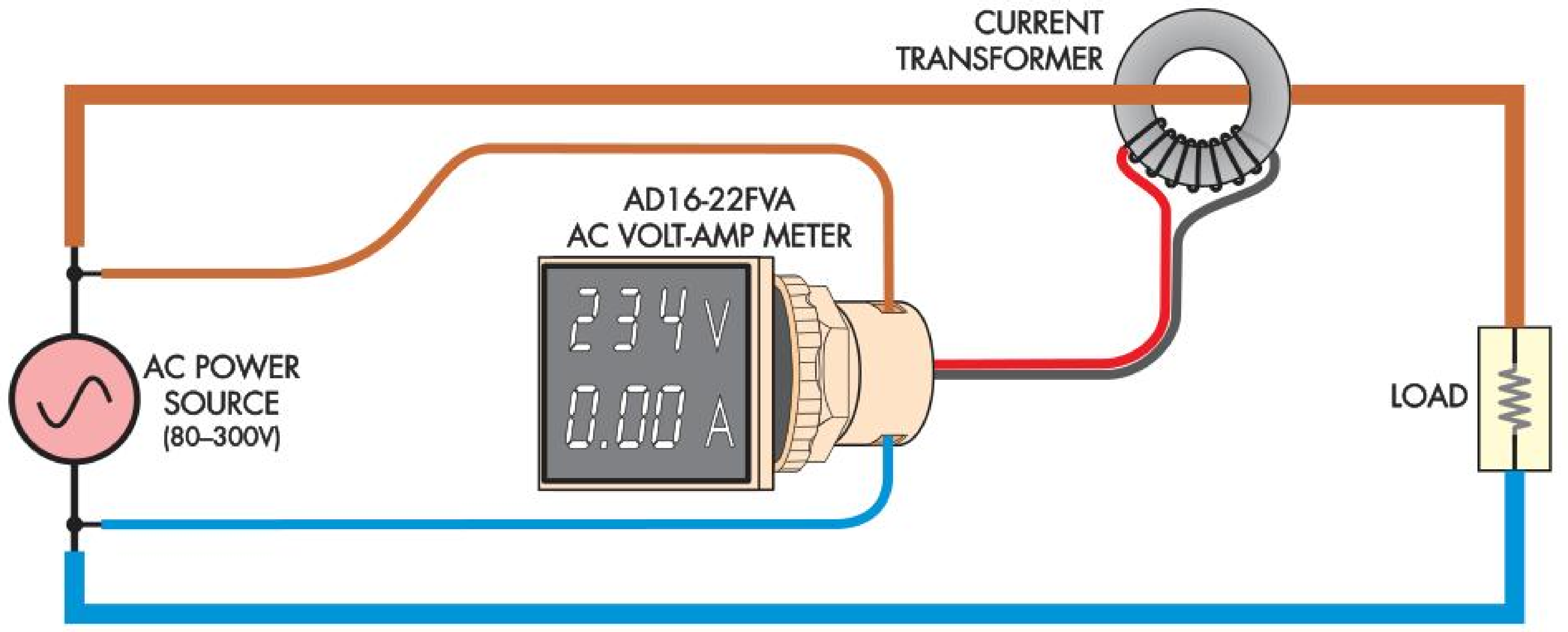
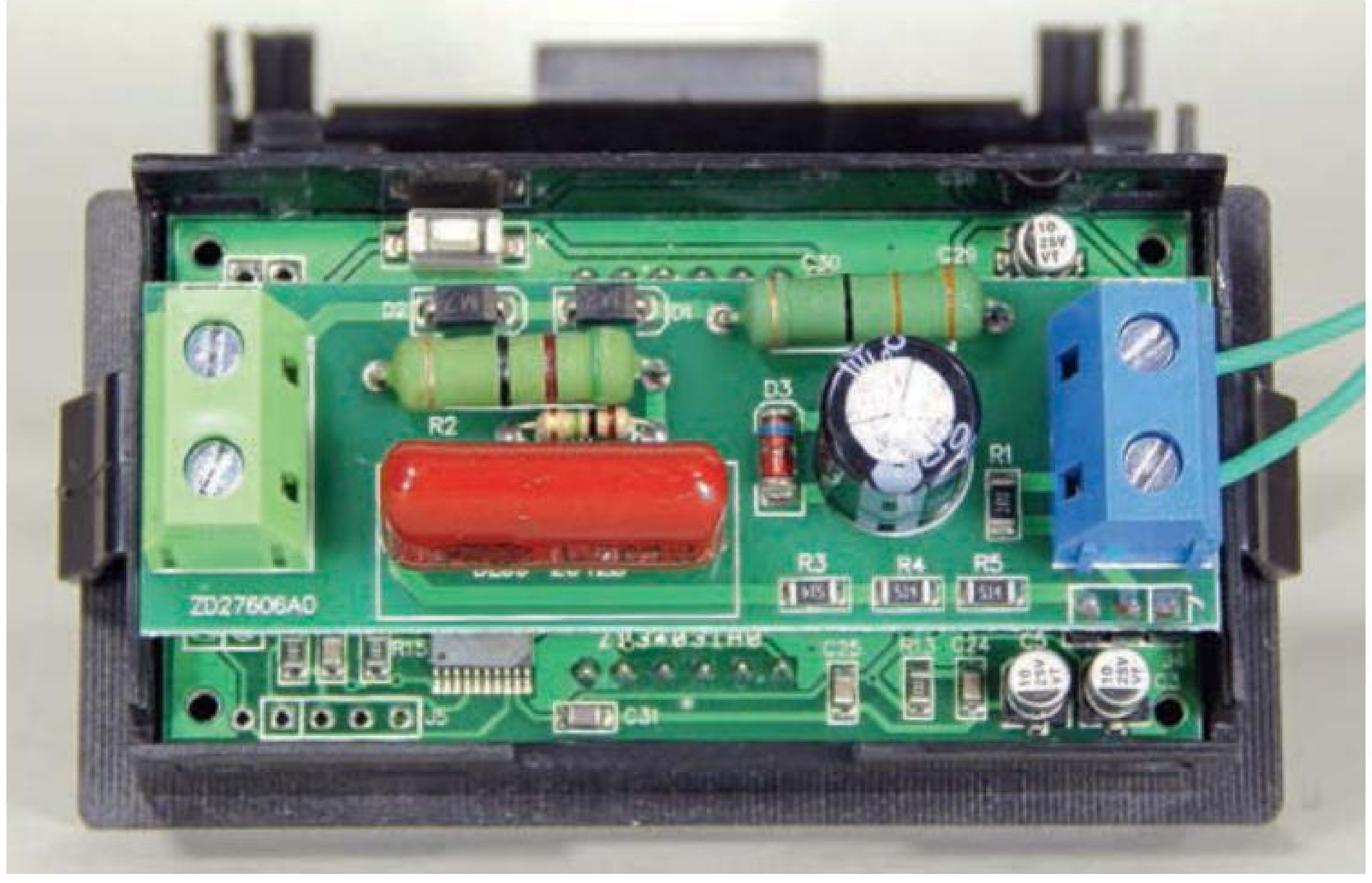


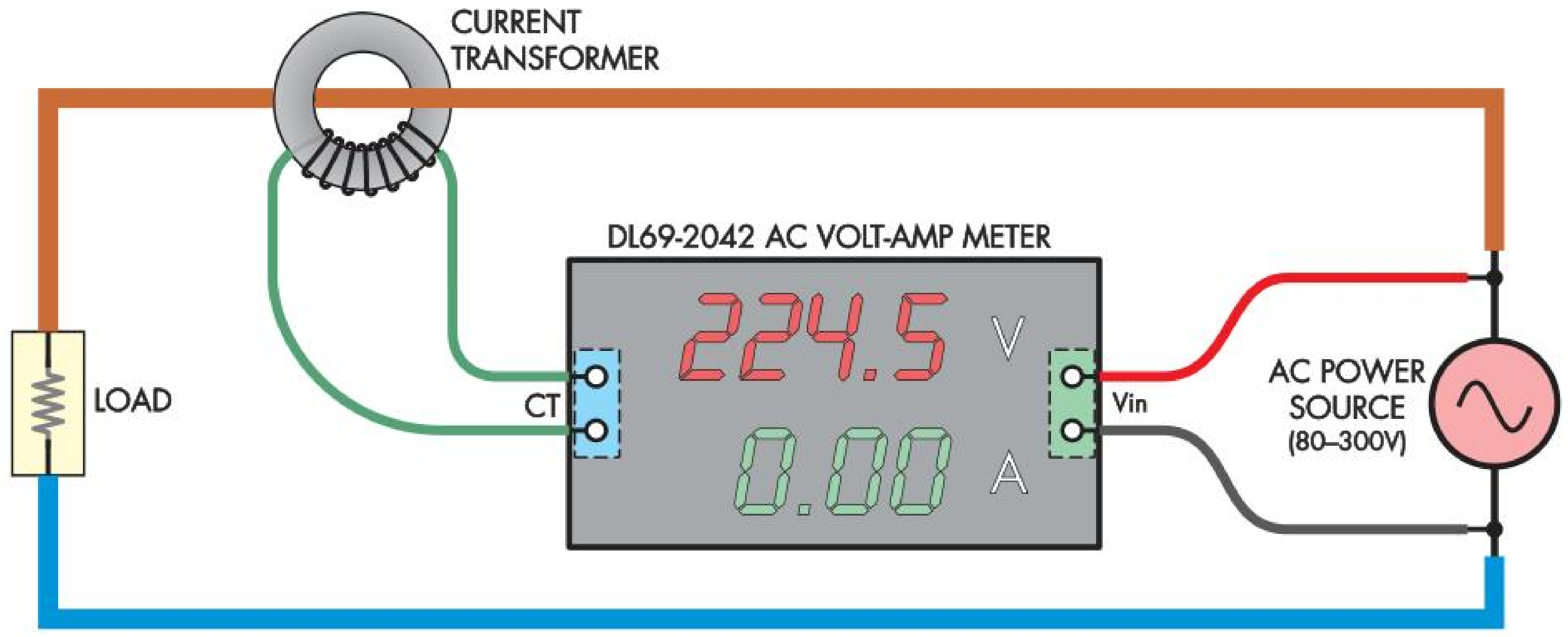
Fig.2: the AD16-22FVA meter is easy to use. One of the power leads from the AC source to the load passes through the CT (polarity connections do not matter), while the other two leads connect across the source.

(Above) the DL69-2042 looks nearly identical to the DSN-VC288 shown in the last article.



The DL69-2042 has a measurement range of 80-300V and 0-100A. There are also some versions with a voltage range of 200-450V.

Fig.3: (Below) as you might expect, like all the other panel meters described in this article, the DL69-2042 is very simple to operate.



the price of the AD16-22FVA. It also arrived safely about 30 days later.

When I checked it out using the same test setup as before, the voltage readings were only 0.2% high while the current readings were 2% high. This was just within spec at the current level concerned (about 6A).

Like the AD16-22FVA, the DL69-2042 is quite easy to use, as you can see from Fig.3. Again all you need to do is thread one of the wires connecting to the load through the centre of the CT core, then connect the meter's voltage input terminals to the same source of AC power.

Both the CT and V_{IN} terminal blocks are on the rear of the meter's case; they're only shown on the front in Fig.3 for clarity.

The larger digits make the DL69-2042 significantly easier to read than the AD16-22FVA, while the 4-digit displays provide higher resolution. So this meter is good value for money even at its higher price. If you only need readings for both voltage and current, it is a good choice.

The PZEM-061 meter

If the PZEM-061 AC meter looks a bit familiar, that's because apart from the accompanying CT, it looks almost identical to the PZEM-051 DC meter module we described in December. That's because it is manufactured by the same firm, Ningbo Peacefair Electronic Technology, in China's Zhejiang province.

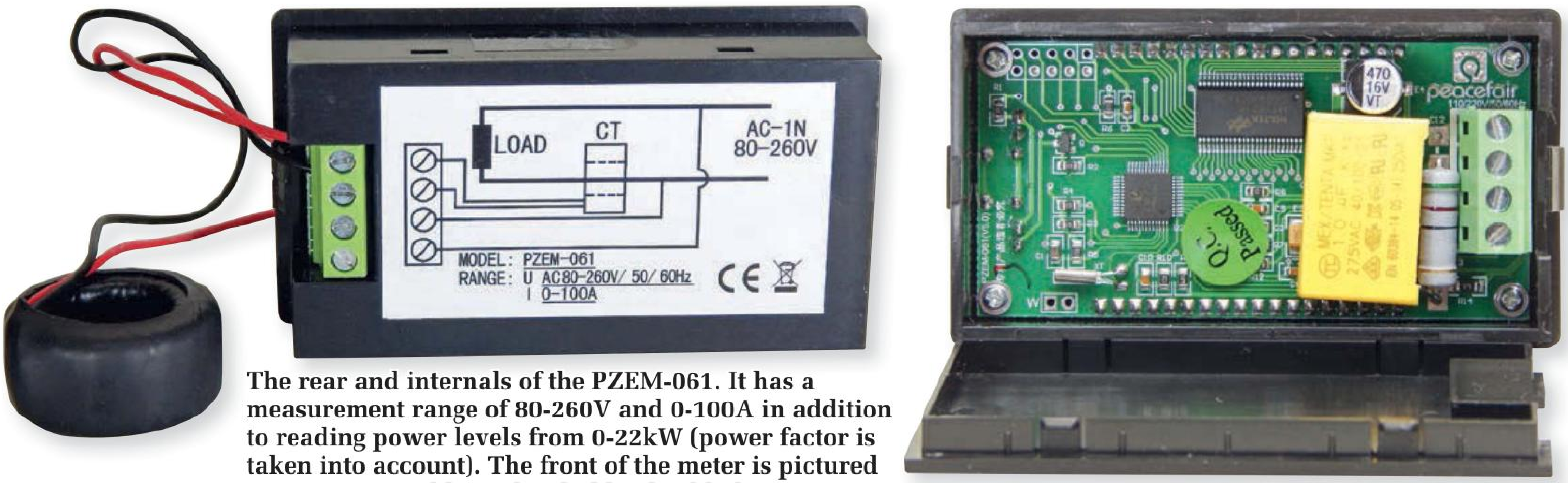
Like the Peacefair DC meter, it comes in a rectangular case measuring 90mm wide, 50mm high and 25mm deep, designed to clip inside a rectangular panel opening 87 x 46mm. Like the DC meter, it also features an LCD window measuring 50 x 30mm with blue LED backlighting, and the main digits are about 6.5mm high.

In addition to the voltage and current readings, it also shows the corresponding power level and energy consumed. All of these parameters are displayed using four digits (see the left-hand photo on the first page).

The voltage measurement range is 80-260V and the current range 0-100A. The power range is 0-22kW, with readings for power levels below 1kW showing as 0.0-999.9W, readings for power levels between 1kW and 10kW showing as 1000-9999W and readings for power levels above 10kW showing as 10.0-22.0kW.

The energy consumed range is 0-9999kWh (kilowatt-hours), with readings below 10kWh showing as 0-9999Wh and readings above 10kWh showing as 10-9999kWh.

It has a small recessed button at centre right on the front panel, allowing you to switch the backlighting on or off, reset the energy consumption level to zero to start a new set of measurements,



The rear and internals of the PZEM-061. It has a measurement range of 80-260V and 0-100A in addition to reading power levels from 0-22kW (power factor is taken into account). The front of the meter is pictured on page 102 and has a bright blue backlight.

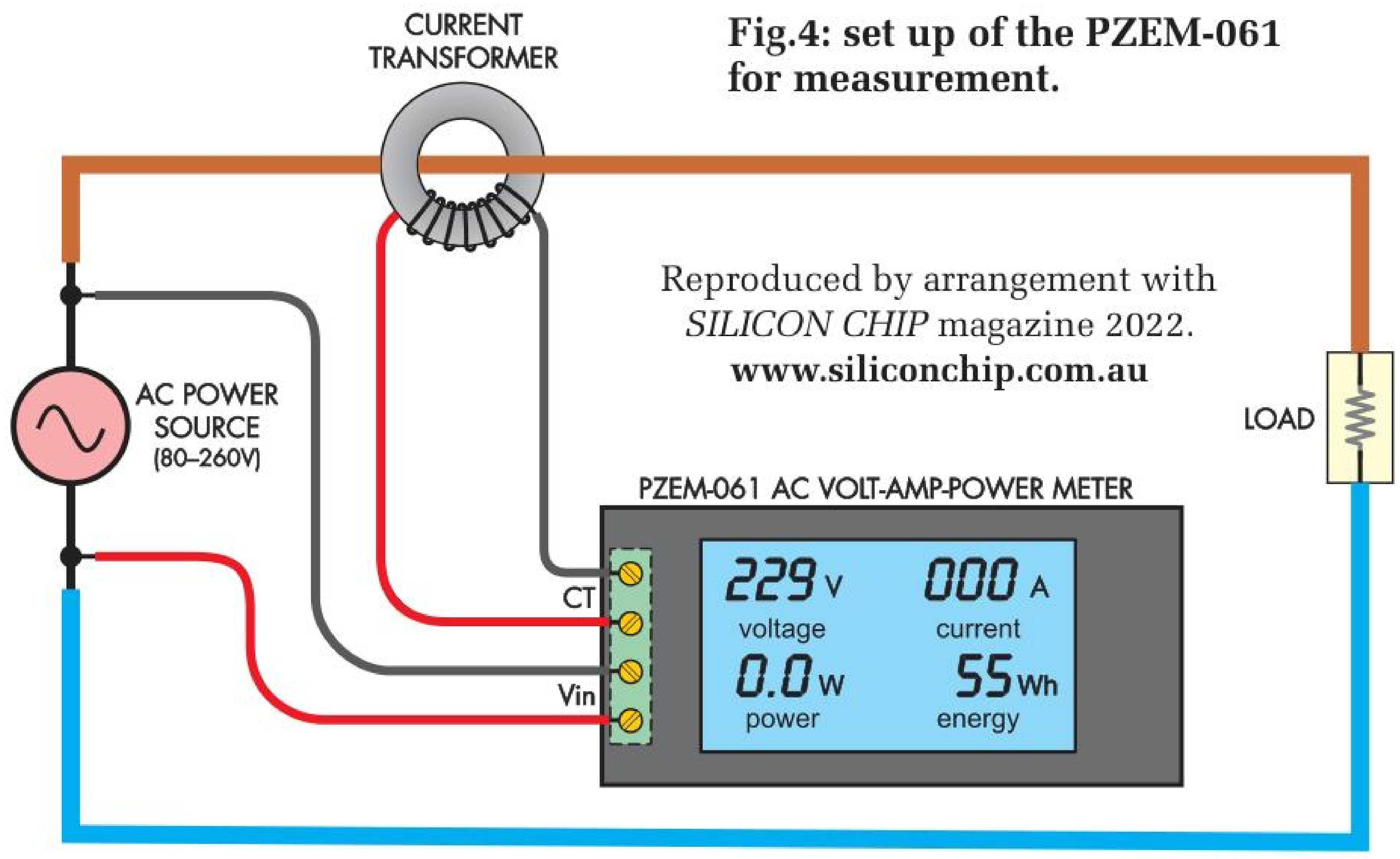


Fig.4: set up of the PZEM-061 for measurement.

Reproduced by arrangement with
SILICON CHIP magazine 2022.
www.siliconchip.com.au

or set a power level alarm threshold to a level between 0.0 and 22.0kW.

The PZEM-061 is again quite easy to use, as you can see from Fig.4. You simply need to pass one of the load power leads through the centre of the CT, and then connect the meter's own power leads to the same source of power. The four-way terminal block is at the rear of the meter, but is shown in Fig.4 at the front, for clarity.

I found the PZEM-061 advertised on the Banggood website for $19.22 plus $3.73 for air parcel delivery. Again, it arrived about 30 days after I ordered it.

The rated measurement accuracy of the PZEM-061 is ±1%, and when I checked it out, I found the voltage readings to be 0.21% high while the current readings were 0.05% high. That is not only well within spec, but quite respectable. The power and energy readings were accurate too; not surprising as these are calculated from the measured voltage and current.

Although the display digits are only 6.5mm high, the blue LED backlighting makes them quite easy to read. So all in all, the PZEM-061 represents excellent value for money.

The D69-2058 meter

The last AC meter we're describing is the D69-2058 multi-function meter. This one is slightly smaller than the PZEM-061 at 80mm wide, 42mm high and 47mm deep, but it displays a total of six measurement parameters: voltage, current, power, mains frequency, energy consumption and power factor (see the right-hand photo on the first page).

The D69-2058 has an LCD screen with digits about 6.5mm high, and it is quite bright, so all the parameters are easy to read. The voltage display has four digits and covers the range of 80-300V (although the meter can alternatively be ordered with a range of 200-450V).

The current range covers 0-99.99A, with a minimum resolution of 0.01A. Power can be displayed over the range 0-9999.9W, with a claimed accuracy of 0.1W. Mains frequency can be displayed over the range 45-65Hz, which should cover all countries outside of odd situations.

Energy consumption can be calculated and displayed over the range from 0-999999kWh, with a resolution of 0.01kWh for values below 1000kWh, a resolution of 0.1kWh for values up to 9999.9kWh, and 1kWh for values up to 999,999kWh. Finally, the power factor is shown as 0.00-1.00.

The rated accuracy of the D69-2058 for voltage and current is ±1%, ±2 LSDs (least-significant digits).

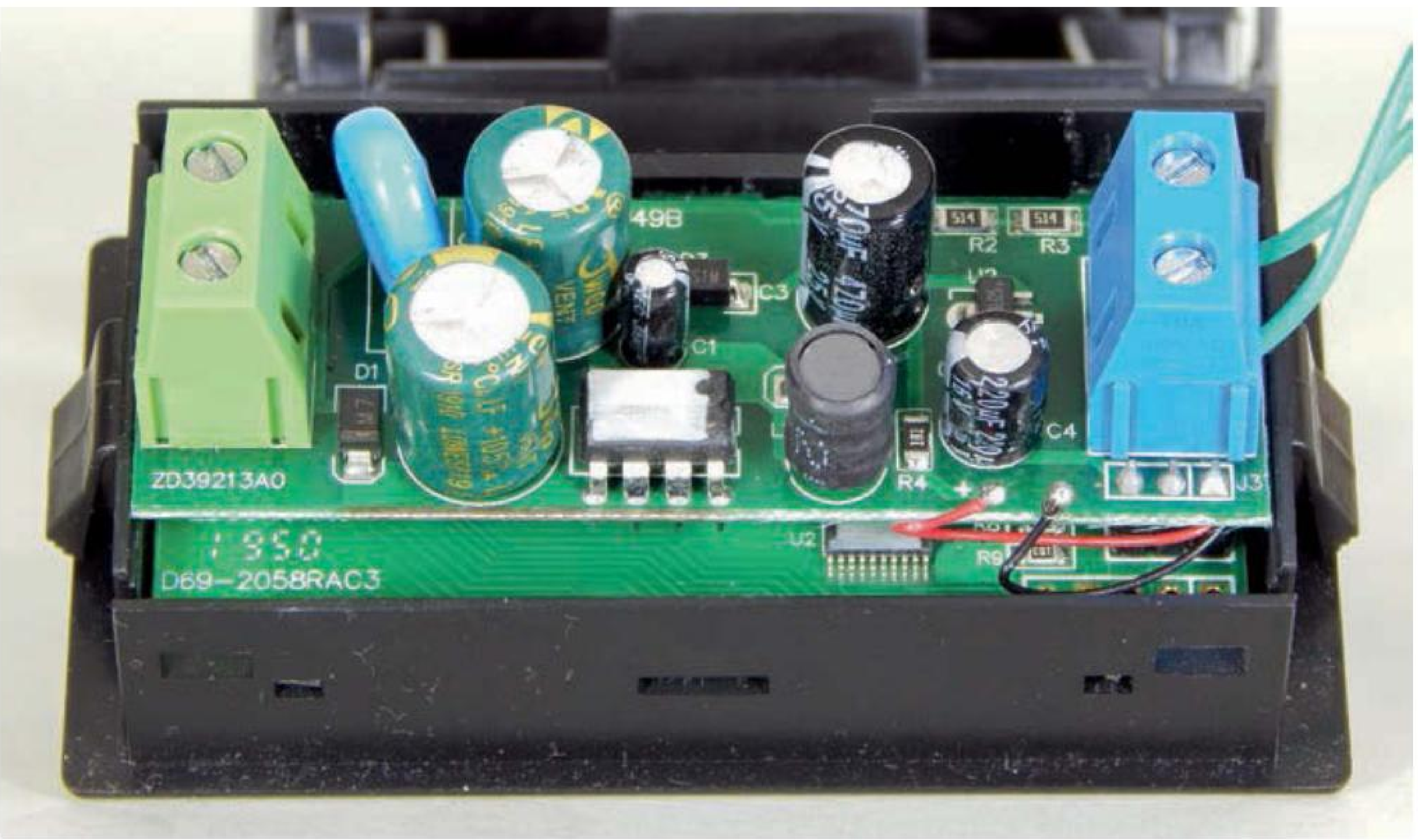
I found the D69-2058 on offer at AliExpress from a supplier called Cooperate Electric Store, for $19.65 plus 81¢ for airmail shipping. It arrived in good condition about 40 days later.

The D69-2058 is just as easy to use as each of the other AC meters, as you can see from Fig.5. I found that the voltage readings were 0.22% high, while the current readings were 0.22% low. So the power readings should be very close to spot-on.

Summary

All of these AC panel meters work well and offer excellent value for money, but the one that impressed me the most was the D69-2058, which not only has the largest number of measurement parameters, but also the most readable display. So, if you need a multi-function AC meter for checking the operation of household appliances or workshop machines, it would make an excellent choice.

It's important to make sure that, regardless of what meter you use, all your mains wiring is properly insulated, and the meter is housed in an appropriate, sturdy case.



Shown above are the internals of the D69-2058 AC panel meter. Compared to the previous three meters, this one offers a lot more features, displaying voltage, current, power, mains frequency, energy consumption and power factor. The front view can be seen on the first page of this article.

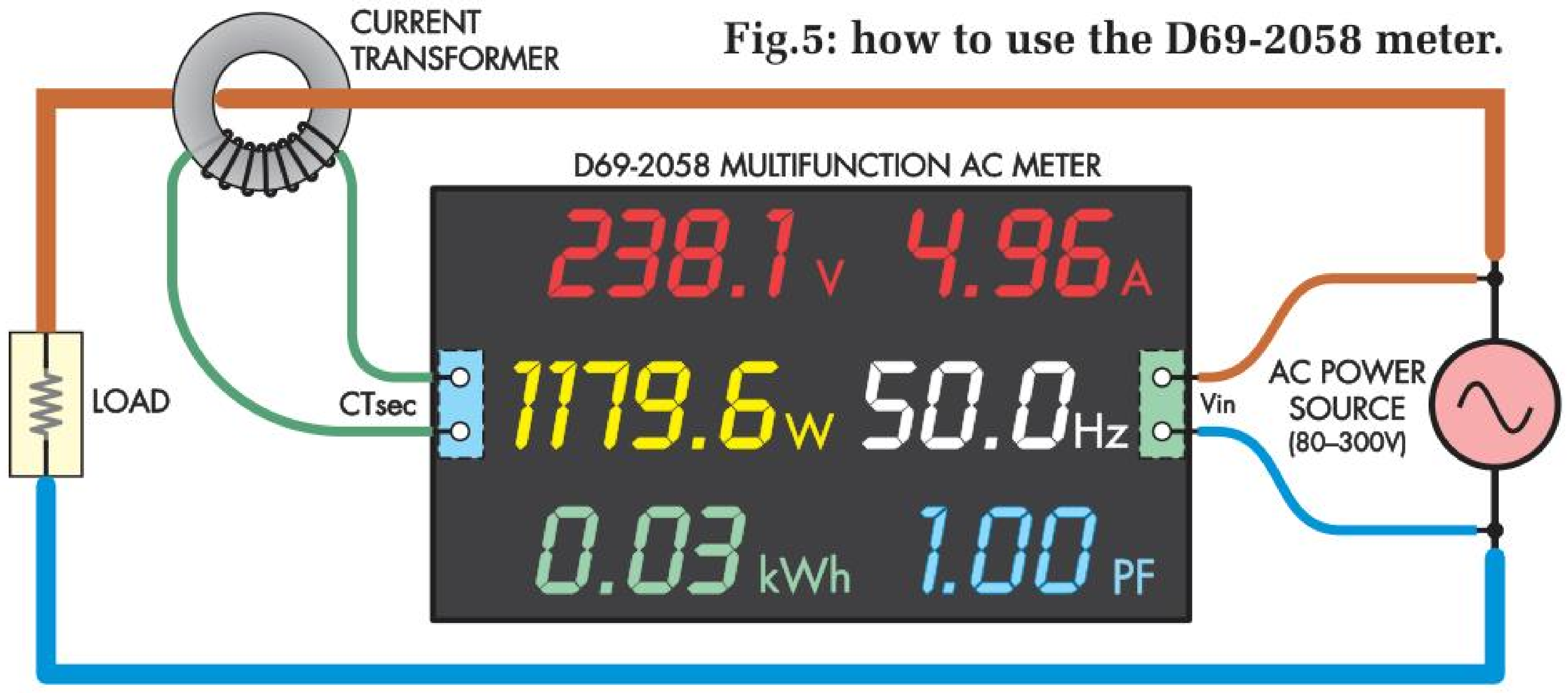


Fig.5: how to use the D69-2058 meter.

KickStart

by Mike Tooley

Part 7: Plug and play with I^2C

Our occasional *KickStart* series aims to show readers how to use readily available low-cost components and devices to solve a wide range of common problems in the shortest possible time. Each of the examples and projects can be completed in no more than a couple of hours using 'off-the-shelf' parts. As well as briefly explaining the underlying principles and technology used, the series will provide you with a variety of representative solutions and examples, along with just enough information to be able to adapt and extend them for your own use.

This seventh instalment provides an introduction to the popular and simple I^2C interface. To help you get started, we've provided a useful practical example in the form of an Arduino Nano-based FM radio using I^2C for controlling both the radio module and an OLED station display.

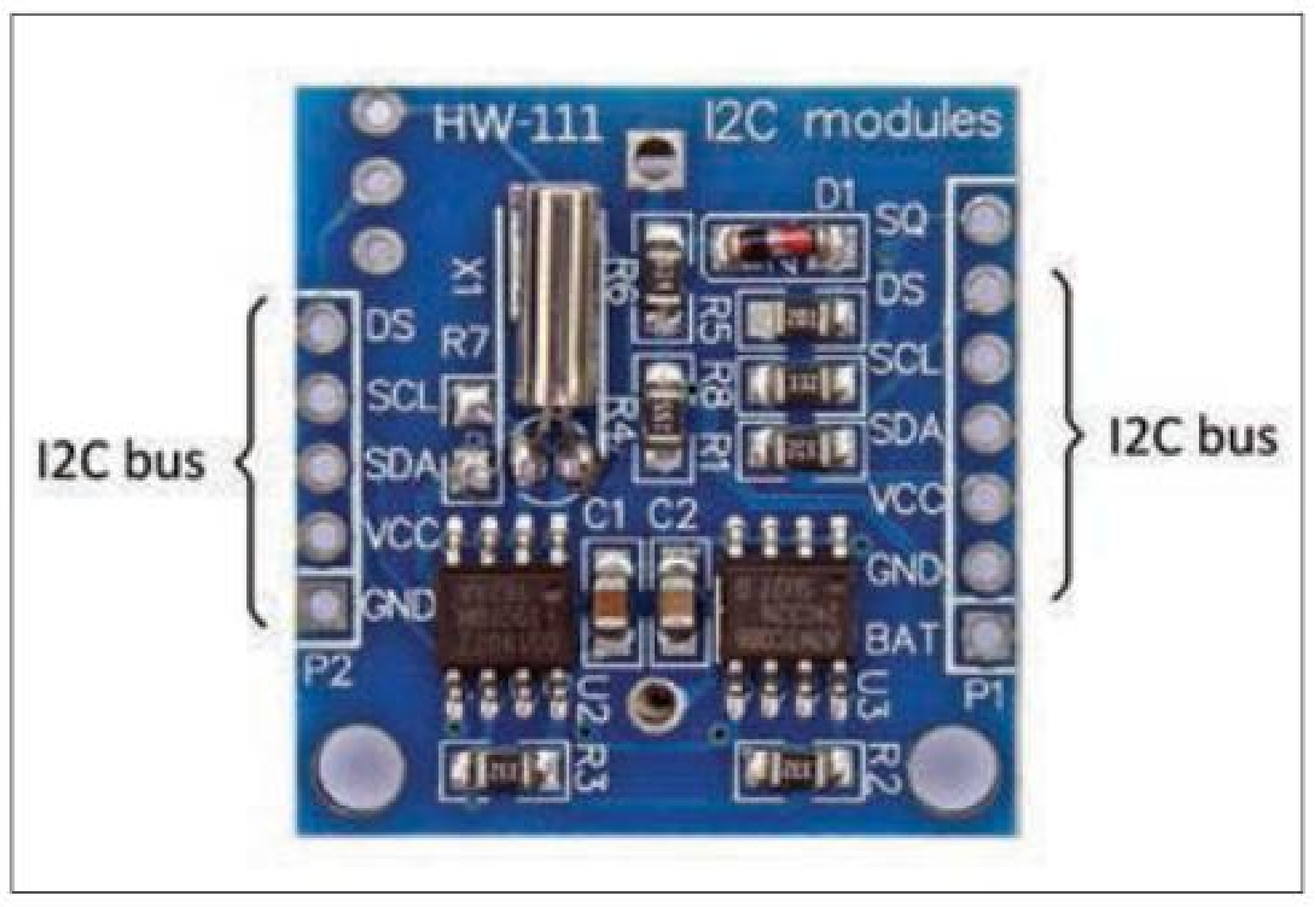


Fig.7.1. A low-cost real-time clock (RTC) designed for use with a wide range of I^2C-compatible microcomputer and microcontroller systems.

Nowadays, microcontrollers and single-board computers usually provide you with a very handy method of connecting external devices using a popular 'two-wire' *inter-integrated circuit* interface supported by a wide range of low-cost bus-compatible devices. This versatile interface is variously known as 'IIC', 'I2C', or 'I^2C', and it will allow you easily to interface your controller with a host of devices, such as I/O multiplexers; sensors for temperature, pressure and humidity; magnetometers; real-time clocks; motion sensors and a variety of display controllers. I^2C is also found in some interesting programmable devices, such as the FM radio chip featured in this article.

I^2C is a very simple bus system where bidirectional serial data appears on one line (SDA) and a clock signal is sent on a second line (SCL). It thus requires only two bus lines plus, of course, a common ground connection. In order to avoid conflict, each device connected to the I^2C bus is software addressable using a unique address. The advantage of these minimal connecting requirements is that equipment based on I^2C can be very easily modified and expanded without the need for major hardware changes.

I^2C was the brainchild of Philips, but several of its leading competitors (including Motorola/Freescale, NEC, Siemens, STM and Texas Instruments) have developed their own I^2C-compatible products. In addition, Intel's SMBus provides a stricter definition of I^2C that helps to improve the interoperability of I^2C devices from different manufacturers.

Fig.7.1 shows a typical I^2C device, a battery-backed real-time clock (RTC) module. I^2C bus compatibility makes it very simple to add this device to a wide range of microcomputer and microcontroller systems. Note that in order to simplify interconnection the I^2C bus connections are duplicated on opposite edges of the board.

Key features of I^2C

Key features of the I^2C bus include:

- Only two bus lines (plus ground) thus minimal interconnecting wiring
- Each device connected to the I^2C bus is software addressable with a unique address (see Table 7.1 for some examples)
- Simple master/slave relationships exist at all times, with bus masters operating as master-transmitters or as master-receivers
- I^2C is a true multi-master bus, which incorporates collision detection and arbitration to prevent data corruption if two or more masters simultaneously attempt to initiate data transfer
- Serial, 8-bit-oriented, bidirectional data transfers can be made at up to 100 kbit/s in standard mode, and up to 400 kbit/s in fast mode. This is perfectly adequate for most non-critical applications

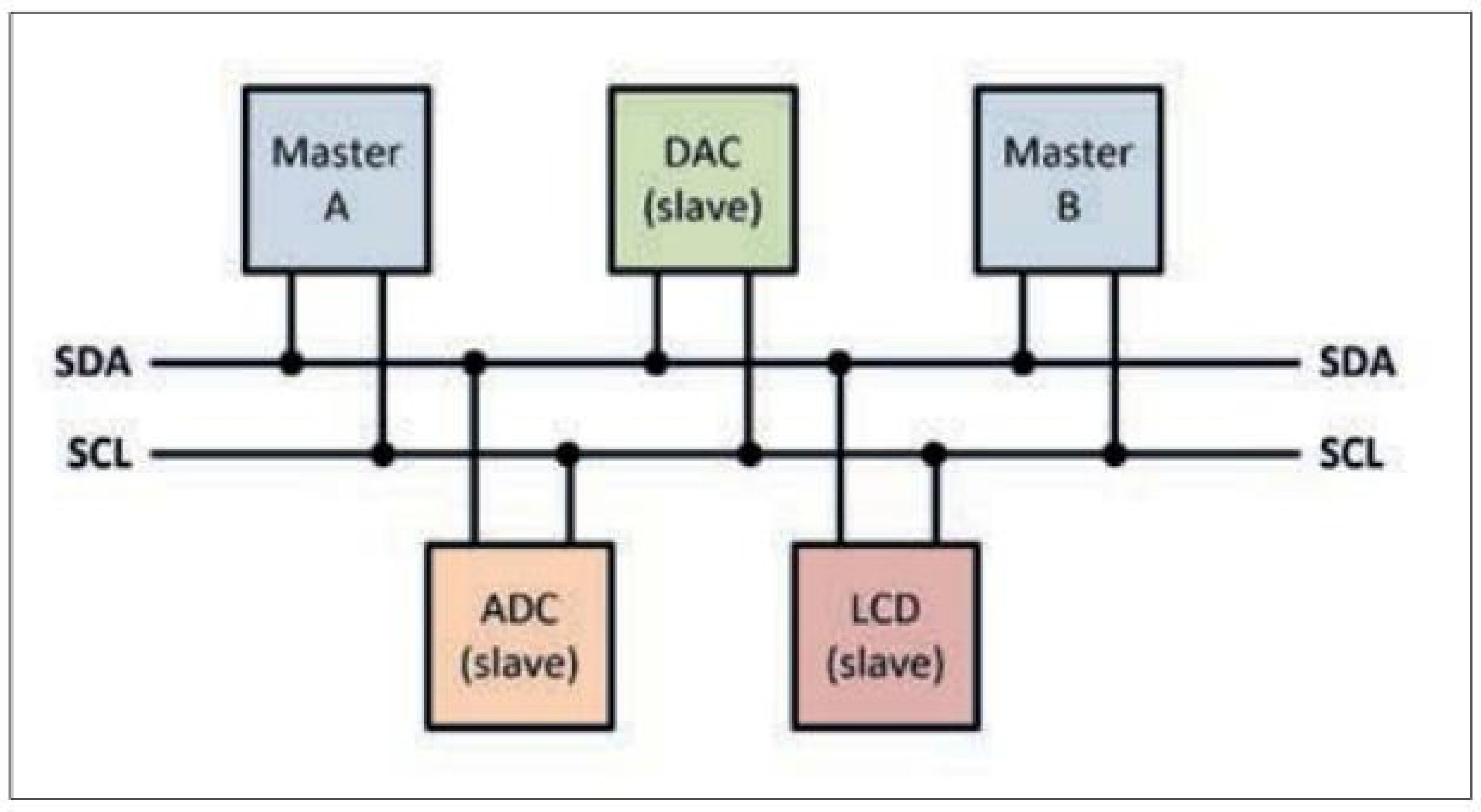


Fig.7.2. I^2C bus with two bus masters and three slaves.

- On-chip filtering is usually incorporated to reject transient noise spikes that can be present on the bus data line
- The number of devices that can be connected to the same bus is limited only by the maximum specified bus capacitance (400pF). This makes I^2C highly expandable.

Bus interface logic

The single data line is shared between multiple devices, so I^2C uses a system of addressing to identify the device that it needs to communicate with. Data communication is initiated by means of a unique start sequence. This involves pulling the data line (SDA) low while the clock line (SCL) is high. This can be achieved by using very simple bus interface logic where each of the bus lines are normally pulled high and driven low when activated by a device connected to the bus (see Fig.7.2).

Fig.7.3 shows a simple bus transaction which begins with a start condition (S) and ends with a stop condition (P). Note

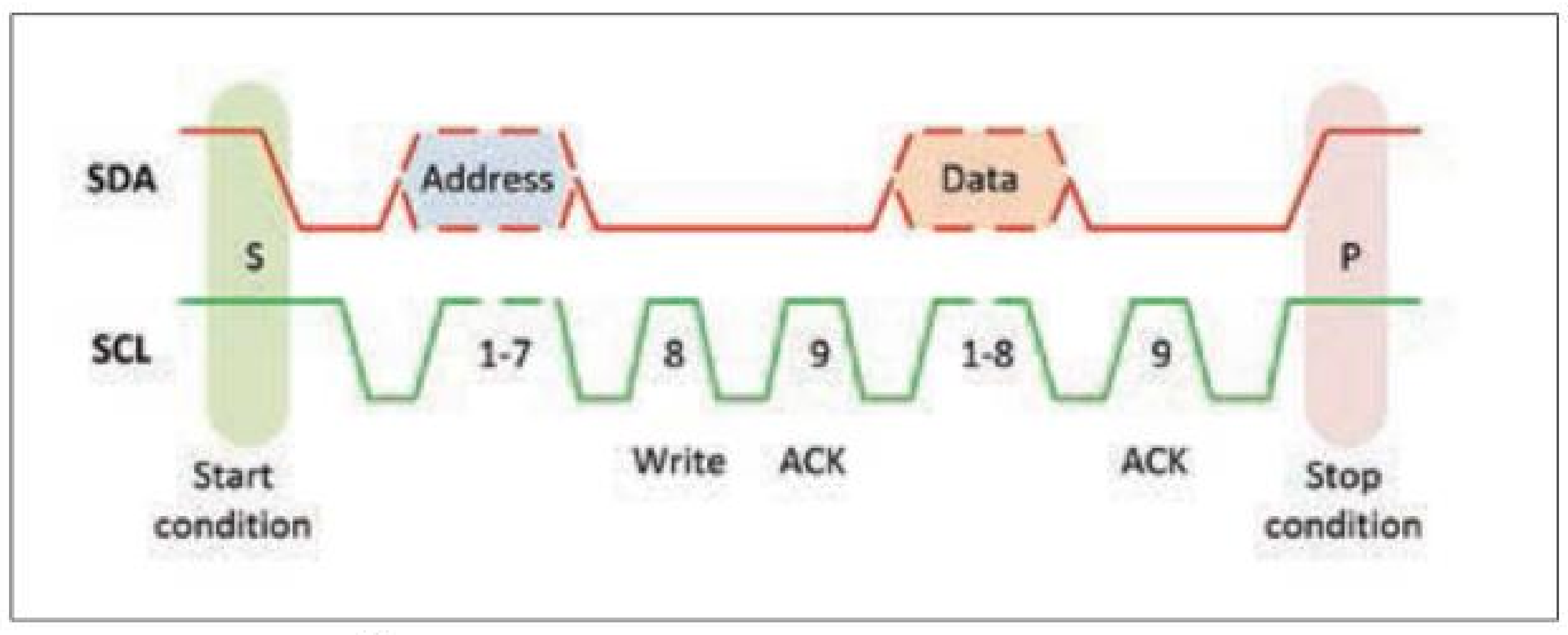


Fig.7.3. An I^2C bus transaction.

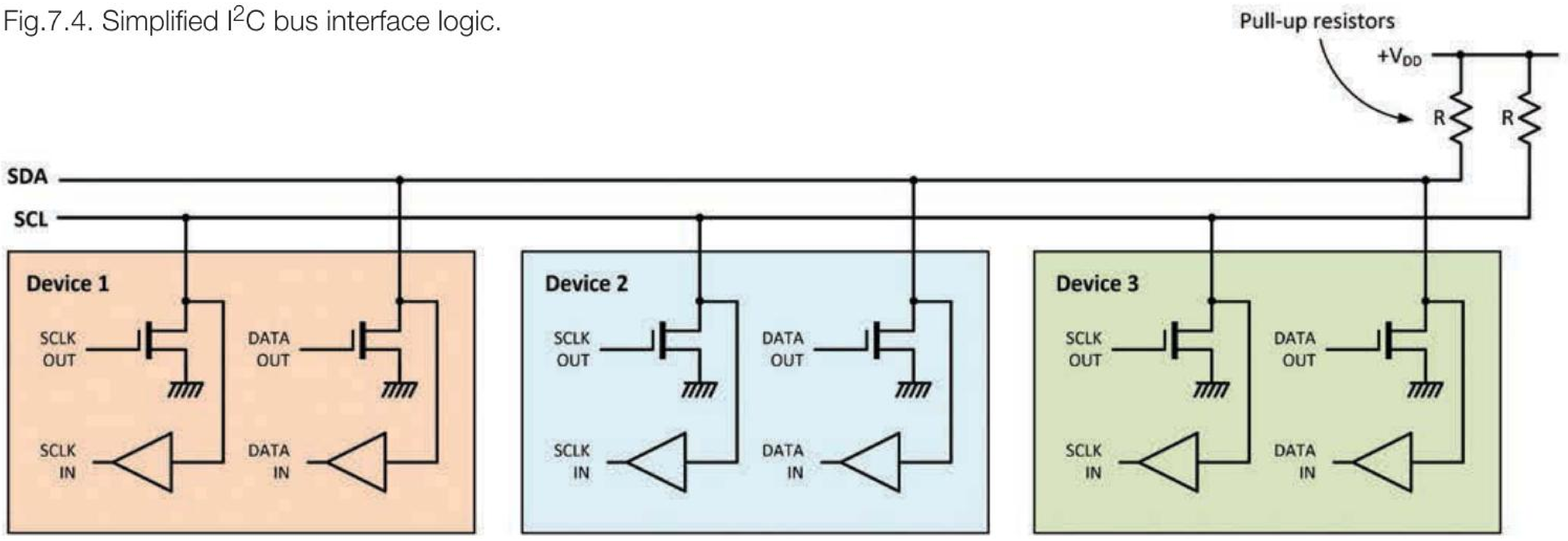


Fig.7.4. Simplified I^{2}C bus interface logic.

how address information is transmitted serially on the SDA line during clock cycles 1 to 7 of the first bus cycle, while serial data follows during clock cycles 1 to 8 of the next bus cycle. Notice also how the bus lines are placed in a quiescent high state both before and after the bus transaction.

Fig.7.4 shows the minimal logic required to interface with the I^{2}C bus. A bidirectional gate arrangement is used for input to and output from both the local serial clock (SCLK) and serial DATA lines of each bus-connected device. Incoming data and clock signals are regularised by means of a high-impedance input buffer stage, while output data and clock signals drive the bus using an open-drain MOS device. Note that the bus requires pull-up resistors so that the bus lines go high before the start condition (S) and after the end condition (P) – as shown in Fig.7.3.

Addressing

Following the start sequence, transmitted data is only allowed to change when the clock is in its low state. In its basic form, and by virtue of the seven bits available for addressing (see Fig.7.3), the I^{2}C protocol caters for a total of 127 devices. In addition to the seven bits used for addressing, the first byte of an I^{2}C transfer generated by a bus master includes a bit that indicates the direction of the data transfer. Note that the address is transferred with the most-significant bit first. Table 7.1 shows I^{2}C addresses for a diverse selection of I^{2}C devices; it should give you an idea of just how versatile and useful this simple bus really is. Note that addresses are quoted in hexadecimal (thus ‘0x20’ is 20 in hexadecimal, 32 in denary or 100000 in binary).

I^{2}C compared with SPI

I^{2}C is a very simple bus system where bidirectional data appears on a single line (SDA) and a clock signal is sent on a second bus line (SCL). Serial peripheral interface (SPI), on the other hand, offers a full-duplex point-to-point connection where the data is passed in and out on *separate* lines (MOSI and MISO). SPI is therefore faster and often easier to use than I^{2}C, but there can often be situations in which I^{2}C is preferred simply because this is the interface that’s built into the chip or device that you intend to use!

Table 7.1: I^{2}C addresses for a selection of devices

Device	Function/application	Address range
BME280	Temp, pressure and humidity sensor	0x76 or 0x77
CAP1188	8-channel capacitive touch sensor	0x28 to 0x2D
MCP23008	I^{2}C GPIO expander	0x20 to 0x27
MCP9808	Digital temperature sensor	0x18 to 0x1F
PCA9685	16-channel PWM driver	0x40 to 0x7F
SAA2502	MPEG audio source decoder	0x30 and 0x31
SSD1306	OLED display driver	0x37
TDA9860	Hi-Fi audio processor	0x40 to 0x41
TEA5767	FM radio receiver	0x60
TMP007	IR temperature sensor	0x40 to 0x47

Introducing the TEA5767 radio chip

The TEA5767 is a single-chip (see Fig.7.5) electronically tuned FM stereo radio designed specifically for use in simple low-voltage applications controlled via the I^{2}C bus. The device is completely adjustment-free and only requires a minimum of small and low-cost external components. The main features of the radio are:

- Integrated RF amplifier for high sensitivity
- 87.5MHz to 108MHz tuning range for the US and Europe (76MHz to 91MHz in Japan)
- RF automatic gain control (AGC)
- Fully integrated FM demodulator
- FM IF selectivity performed internally
- Crystal reference frequency oscillator (operating at 32.768kHz or at 13MHz crystal with an externally applied 6.5MHz reference)
- Synthesised phase-locked loop (PLL) tuning
- Soft mute and signal-dependent mono-to-stereo switching.

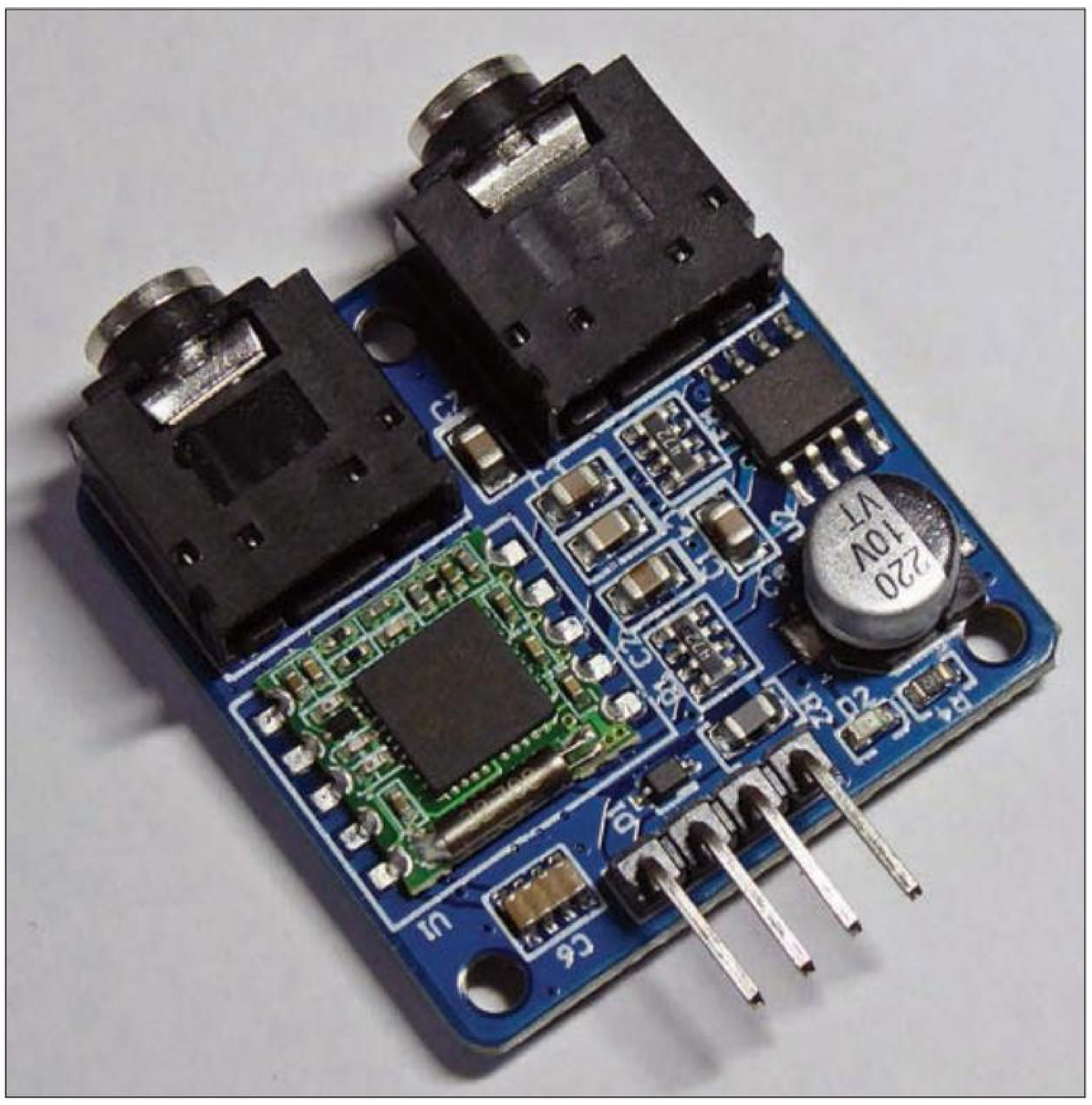


Fig.7.5. The TEA5767 FM radio module.

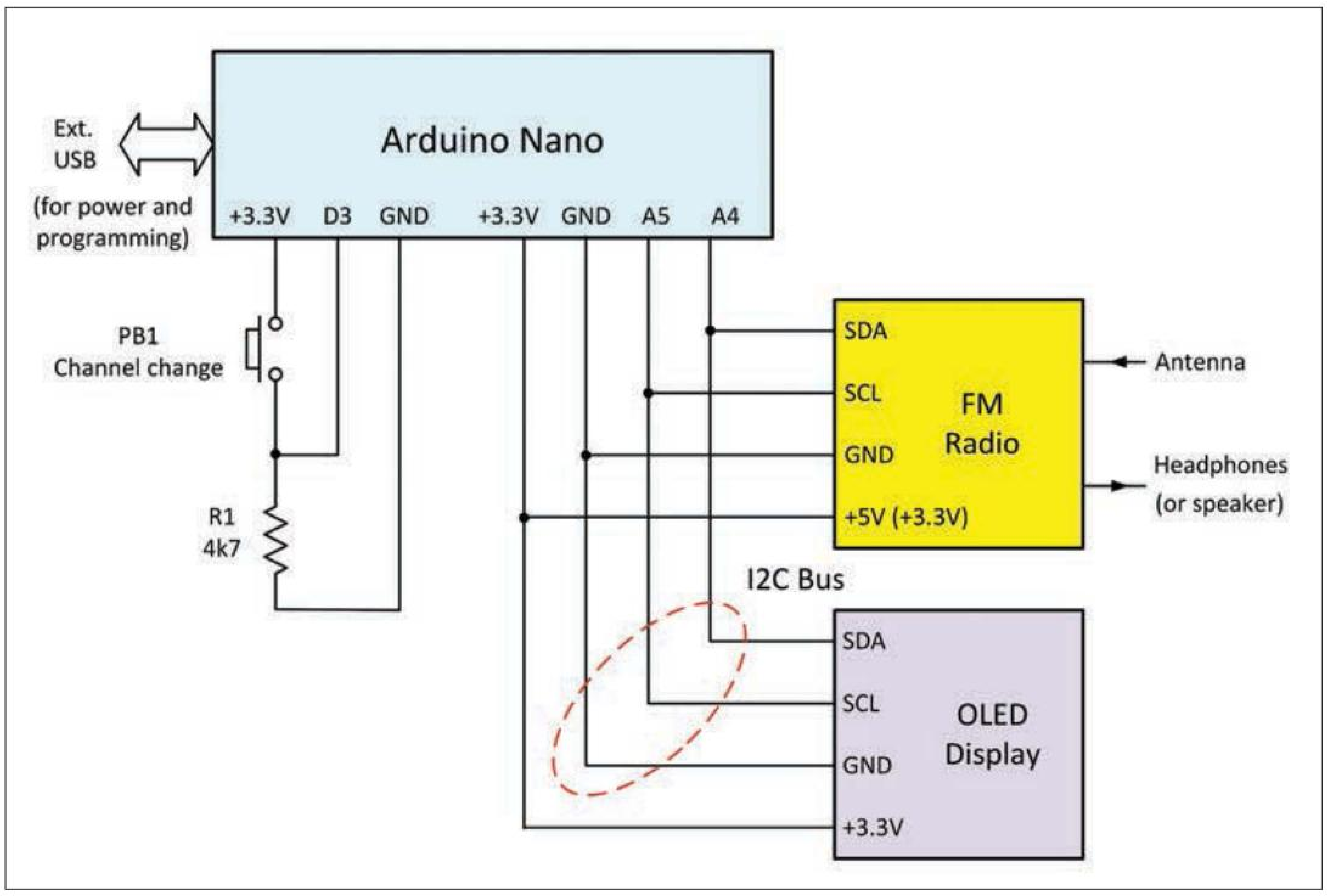


Fig.7.6. Circuit of the I^{2}C-controlled FM radio.

The TEA5767 is easy to use and only requires a few lines of code for channel selection. For example, the following single line of code sets the TEA5767's operating frequency to 104.8MHz (BBC Radio Sussex):

```
radio.setFrequency(104.8); // BBC Radio Sussex
```

Note that the appropriate library routines must first be referenced for inclusion in your code or an error message will be generated and the application will simply not run.

OLED display

The 0.91-inch OLED (organic light-emitting diode) display uses a matrix of 128x32-pixel LEDs. The device is controlled using an SSD1306 driver chip, which also contains an I^{2}C interface. The I^{2}C address of the device is 0x3C and it operates from a supply voltage in the range 3.3 to 5V at a typical current of less than 8mA. The OLED display is very easy to use and only requires a few lines of code. For example, the following code fragment displays a simple text message:

```
u8g2.clearBuffer(); // First clear the display memory
u8g2.setFont(u8g2_font_helvB14_tf);    // Select
the font to be used
u8g2.drawStr(0,25,"BBC Radio Sussex"); // The text
to be displayed
u8g2.sendBuffer(); // Finally send it to the display
```

Note once again that the necessary library routines must be referenced for inclusion in the code and the display driver must first be initialised.

The I^{2}C-controlled FM radio

The circuit of our I^{2}C-controlled FM radio is shown in Fig.7.6. It shows the Arduino Nano controller linked to the TEA5767 FM radio and OLED display using the I^{2}C bus (the SDA and SCL lines are respectively connected to pins A4 and A5 on the Nano).

The code for the circuit (**I2C_FM_radio.ino**) is available for download from the February 2022 page of the *PE* website.

The channel change button (PB1) is connected to digital input D3 on the Nano. This line is configured as an input using the following two lines of code:

```
const int buttonPin = 3;    // Channel change button
pinMode(buttonPin, INPUT);
```

The channel selector (change) button (PB1) is repeatedly pushed to cycle through the four channels and its state (HIGH or LOW) is stored in the `buttonState` variable.

The operating frequencies and channels shown in the code are for use in the Sussex area of the UK, but they can be easily changed to those being used in your local area. All you need to do is change the frequency and station text for each channel. If required, other stations can be catered for by adding further code after the block for Channel 4. A few of the most popular stations are listed in Table 7.2, and most UK readers should be within range of one or more of these. Note that it will be necessary to alter the code at the top of the loop to reflect the new number of available channels.

Constructing the Arduino-based TEA5767 FM radio

The minimal wiring for the Arduino-based FM radio is shown in Fig.7.7. The three modules (Arduino Nano, TEA5767 and OLED display) can be conveniently mounted in a small ABS enclosure, allowing access to the two 3.5mm jack sockets used

Table 7.2: Some popular UK FM radio stations

Station	Frequency (MHz)
BBC Radio Bristol	94.9 and 104.6
BBC Radio Cornwall	95.2 and 103.9
BBC Radio Cymru	92.4 and 92.7
BBC Essex	95.3 and 103.5
BBC Radio Humberside	95.9
BBC Radio Kent	96.7 and 104.2
BBC London	94.9
BBC Radio Manchester	95.1
BBC Merseyside	95.8
BBC Radio Scotland	92.5, 92.6, 92.7, 92.8, 92.9 etc.
BBC Radio Sheffield	88.6
BBC Radio Solent	96.1 and 103.8
BBC Radio Ulster	93.1
BBC Radio Wales	90.2, 90.3, 90.4, 90.5, 90.4 etc.
BBC Radio York	95.5, 103.7, and 104.3
Belfast 89FM	89.3
Capital FM	97.4, 103.2
Classic FM	100.1, 100.2, 100.3, 100.4 etc.
Greatest Hits Radio	96.2 and 97.4
Heart (Essex, Solent, Surrey, Sussex)	97.5
Heart (North East, West Midlands)	100.7, 101.2
Heart (North Wales)	88.0
Heart (Scotland)	101.1, 103.3
Heart (South West)	100.8, 101.2
Manx Radio	88.9

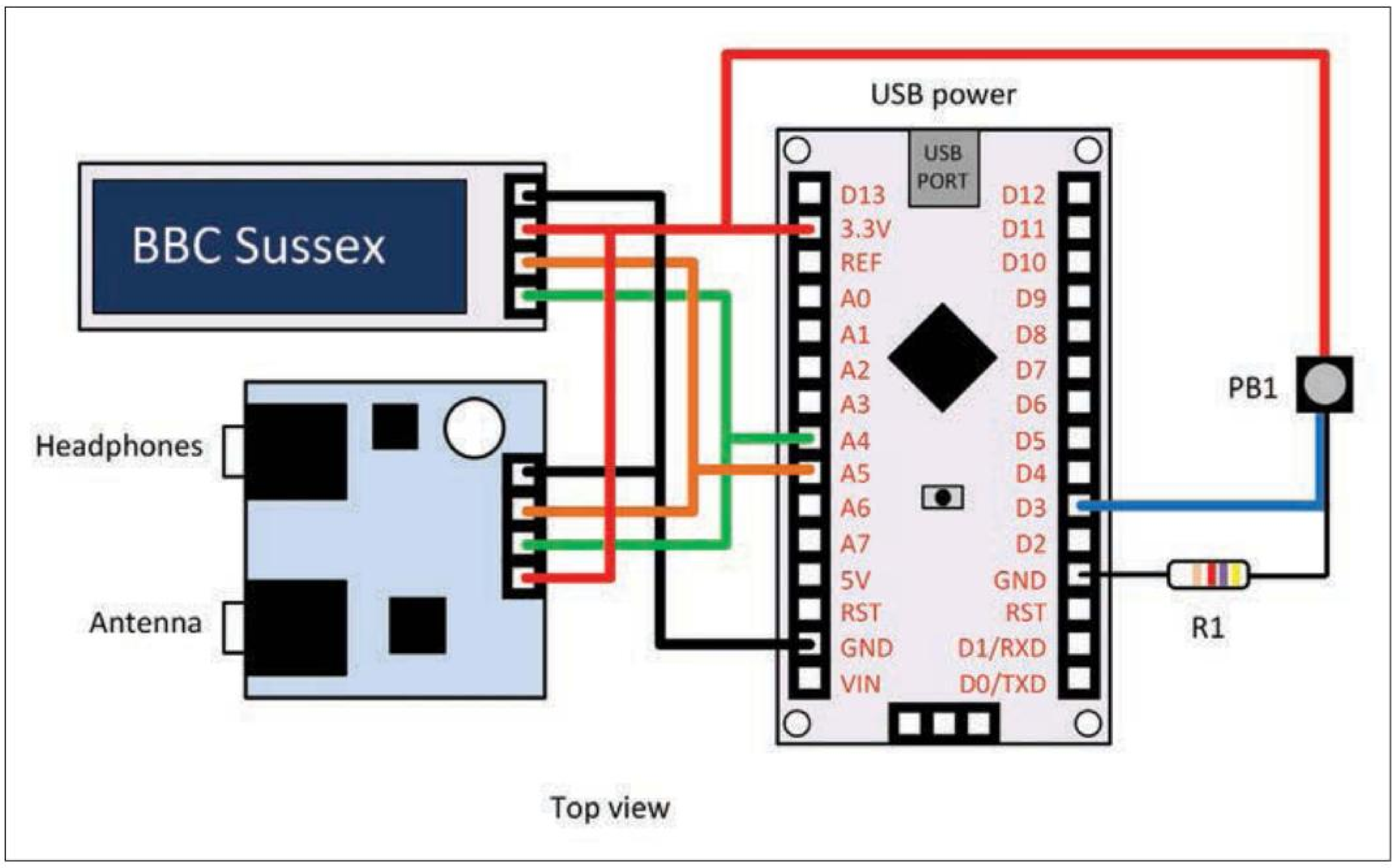


Fig.7.7. Wiring schematic for the I^{2}C-controlled FM radio.

for the antenna input and headphone/speaker output. Access will also be required for the USB programming/5V external DC supply connector. The wiring schematic is shown in Fig.7.7 and a prototype breadboard layout is shown in Fig.7.8.

Finally, this project provides plenty of scope for modification and experimentation. A capacitive touch switch could easily be added for pre-set channel selection. Alternatively, a rotary encoder could be used to give full tuning coverage of the VHF band in 100kHz steps. An Arduino Uno could be substituted for the Nano and the raw DC input could be used instead of the USB connector. The two YouTube video presentations listed in *Going further* overleaf should provide you with plenty of food for thought!

You will need...

To build the I^{2}C radio you need one each of the following:

- Arduino Nano
- I^{2}C-bus-compatible TEA5767 FM radio module (see *Going further*)
- I^{2}C-bus-compatible OLED display module (see *Going further*)
- Telescopic antenna fitted with 3.5mm jack plug (often bundled with the TEA5767 radio module)
- Headphones or external speaker fitted with a stereo 3.5mm jack plug
- Miniature NO (normally open) pushbutton switch
- 4.7kΩ resistor
- USB lead for connection to a PC (for programming)
- USB 5V DC power supply (or any switched 5V supply)
- Small ABS enclosure (or prototype breadboard for test purposes).

Going further

This section details a variety of sources that will help you locate the component parts and further information that will allow you to understand I^{2}C and add a wide range of I^{2}C-bus-compatible devices to your projects. It also provides links to relevant underpinning knowledge and manufacturers' data sheets.

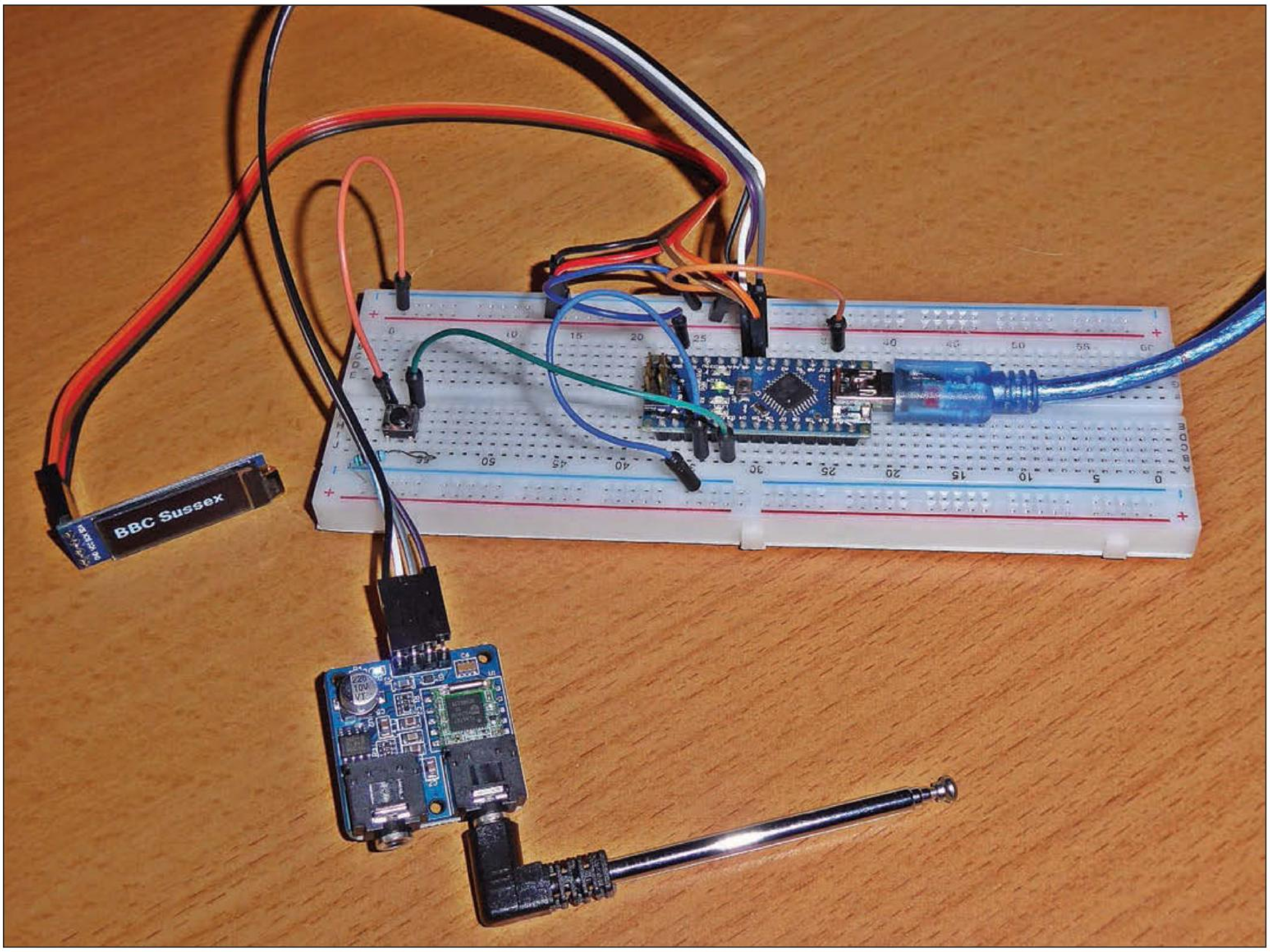


Fig.7.8. The prototype I^{2}C-controlled FM radio on-test, tuned to (FM) BBC Radio Sussex.

Table 7.3: Going Further with I^{2}C

Topic	Source	Notes
Meet the I^{2}C bus	Texas Instruments have a useful introduction to the I^{2}C: **https://bit.ly/pe-Feb22-ks1** For a comprehensive directory of I^{2}C addresses, visit: **https://i2cdevices.org**	An excellent PowerPoint introduction to I^{2}C can be found at: **https://bit.ly/pe-feb22-ks2** To identify the I^{2}C address of a connected device, a scanner application (**i2c_scanner**) is one of the example files in the I^{2}C Wire library. Note: this sketch tests the standard 7-bit addresses and any devices with higher-bit addresses might not be seen properly.
Arduino Nano	The Arduino website provides a variety of resources to support the Nano: **https://bit.ly/pe-dec21-ard1** The Arduino's integrated development environment (IDE) can be downloaded from: **https://bit.ly/pe-feb22-ks3** *Electronics Teach-In 8 – Introducing the Arduino* (available from *Practical Electronics*) provides a one-stop source of ideas and practical information.	An Arduino Uno can be substituted for the Nano, but it will require a much larger enclosure. You may need to use the IDE's in-built Library Manager to download the two library files listed in the code.
TEA5767 FM radio module	The TEA5767 FM radio module is available from several on-line suppliers, including Amazon and eBay. The NXP/Philips datasheet for the chip itself (*not* the complete module) is available from: **https://bit.ly/pe-feb22-ks4**	A more complex FM radio design based on the TEA5767 can be found at: **https://bit.ly/pe-feb22-ks5** This Instructables tutorial provides details of a TEA5767 FM radio based on an Arduino Uno (rather than a Nano): **https://bit.ly/pe-feb22-ks6**
0.91-inch OLED display	The 0.91-inch OLED display using an SSD1306 driver is available from on-line suppliers, including AZ-Delivery, Amazon and eBay – search: '0.91" inch OLED SSD1306'.	AZ-Delivery provide a useful eBook *Quick Start Guide*. Download it for free at: **https://bit.ly/pe-feb22-ks7**
YouTube videos	Csongor Varga has produced a detailed video showing the TEA5767 FM radio module in action: **https://youtu.be/yp0HVGjakMs** Ralph Bacon provides another excellent introduction to using the TEA5767: **https://youtu.be/yWf9uxL6zgE**	These two video presentations differ in style and content, but both will provide you with plenty of useful background information, as well as some useful ideas for further development.

Teach-In 8 CD-ROM Exploring the Arduino

This CD-ROM version of the exciting and popular *Teach-In 8* series has been designed for electronics enthusiasts who want to get to grips with the inexpensive, immensely popular Arduino microcontroller, as well as coding enthusiasts who want to explore hardware and interfacing. *Teach-In 8* provides a one-stop source of ideas and practical information.

The Arduino offers a remarkably effective platform for developing a huge variety of projects; from operating a set of Christmas tree lights to remotely controlling a robotic vehicle wirelessly or via the Internet. *Teach-In 8* is based around a series of practical projects with plenty of information for customisation. The projects can be combined together in many different ways in order to build more complex systems that can be used to solve a wide variety of home automation and environmental monitoring problems. The series includes topics such as RF technology, wireless networking and remote web access.

PRICE £8.99

Includes P&P to UK if ordered direct from us

PLUS: PICs and the PICkit 3 – A beginners guide

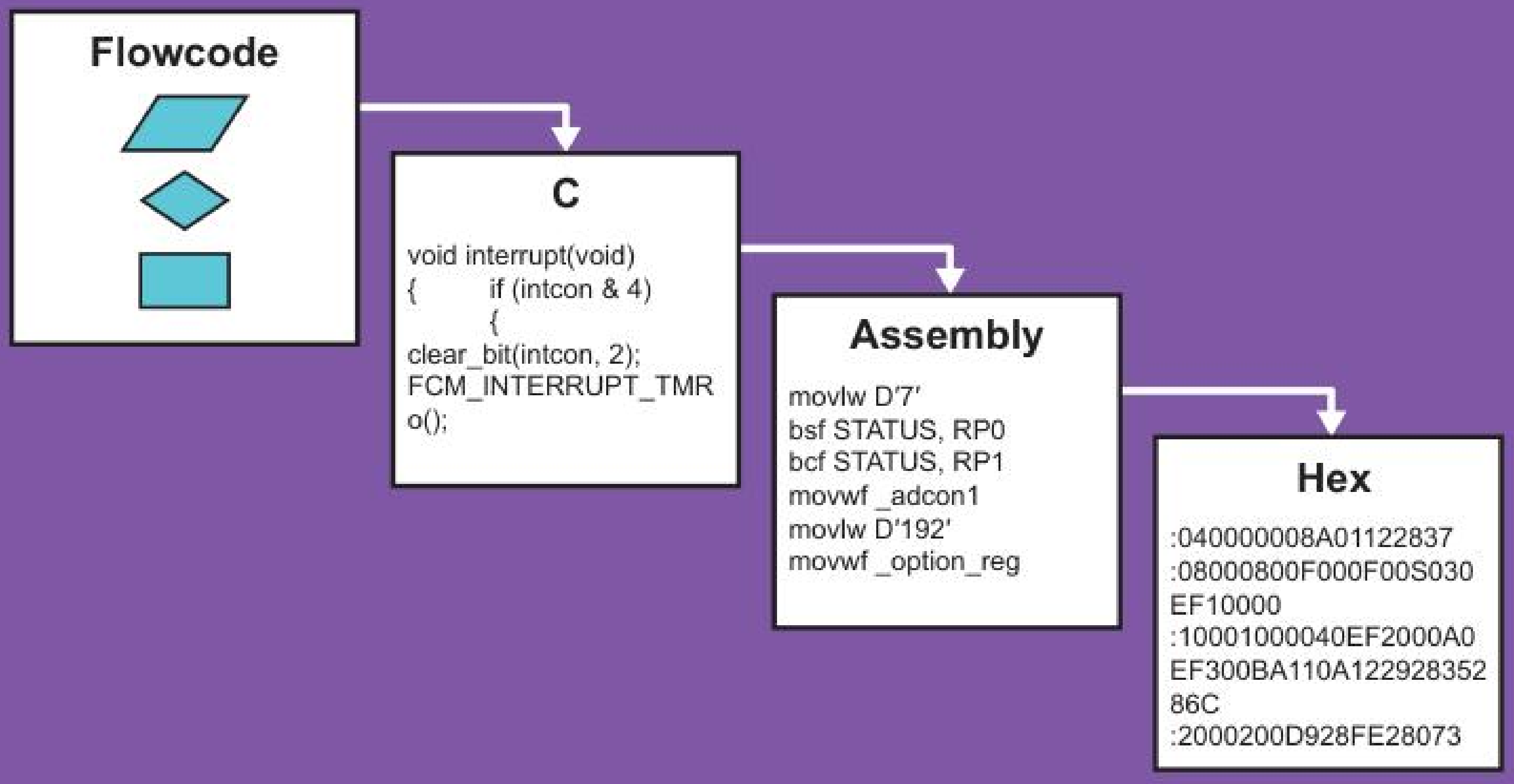
The CD-ROM also includes a bonus – an extra 12-part series based around the popular PIC microcontroller, explaining how to build PIC-based systems.

SOFTWARE

The CD-ROM contains the software for both the *Teach-In 8* and PICkit 3 series.

ORDER YOUR COPY TODAY at: www.electronpublishing.com

Flowcode
Graphical Programming



```
void interrupt(void)
{         if (intcon & 4)
          {
clear_bit(intcon, 2);
FCM_INTERRUPT_TMR
o();
```

```
movlw D'7'
bsf STATUS, RP0
bcf STATUS, RP1
movwf _adcon1
movlw D'192'
movwf _option_reg
```

```
:040000008A01122837
:08000800F000F00S030
EF10000
:10001000040EF2000A0
EF300BA110A122928352
86C
:2000200D928FE28073
```

Programming with Flowcode: RedBoard and PICs – Part 2

Last month, I explained how to test if Flowcode is properly set up to work with a particular choice of target (microcontroller) by creating a simple one-second LED flasher – a simple but important step in building a project. We then started to create a more advanced LED design – a set of traffic lights – which will demonstrate how to use some of the most common functions in Flowcode.

We began by creating a traffic lights graphical simulation element, and now comes the fun part – designing the program and creating the code, but doing this without writing any code in the traditional line-by-line way.

Creating the code

This is our first proper project, so we'll keep it simple. The very first step is to decide what we want our program to do – the algorithm. So, before we go any further let's specify program operation. We want to power up our LED traffic lights hardware and then cycle through the correct lighting sequences, controlled by a single push-button, as follows:

1. Power up circuit, Red on
2. Enter loop, wait for button press
3. Press button
    Red stays on, Amber on
    Delay
    Red off, Amber, off Green on
    Stop until the next button push
4. Press button
    Green off, Amber on
    Delay
    Amber off, Red on
    Stop until the next button push
5. Return to 2.

Item 1 is our turn-on/initialisation step. Cycles 3 and 4 are two sequences that move the traffic lights from stop to go and from go to stop. (Note that the order defined above is for UK traffic lights and may be subtly different to your local traffic lights. If so, just follow along with our choice, and when you are confident you understand the design process amend it to suit your local lights.)

Last month's one-second LED flasher program has now served its purpose, so all the icons within that program may now be deleted and will be replaced with our traffic lights program.

Macros

For all types of coding, whether you are using complex languages like C or graphical systems such as Flowcode, it's good practice to use 'functions'. In Flowcode, functions are called 'User Macros', and we'll be making a lot of use of these important elements.

User Macros unclutter the program, which in Flowcode is called the 'Main Macro', and make it much easier to build and follow. Our Main Macro will include three User Macros called `InitialState`, `Go` and `Stop`. (Do notice the format of the name `InitialState` – it does not use spaces, which are not allowed. However, underscore is valid, so `Initial_State` could be used instead if that is your preferred style.)

I like to use Project Explorer (enabled via View Ribbon) for adding User Macros, Variables, Component Macros or Command Icons – don't worry, all these new concepts will be covered when we need to add them.

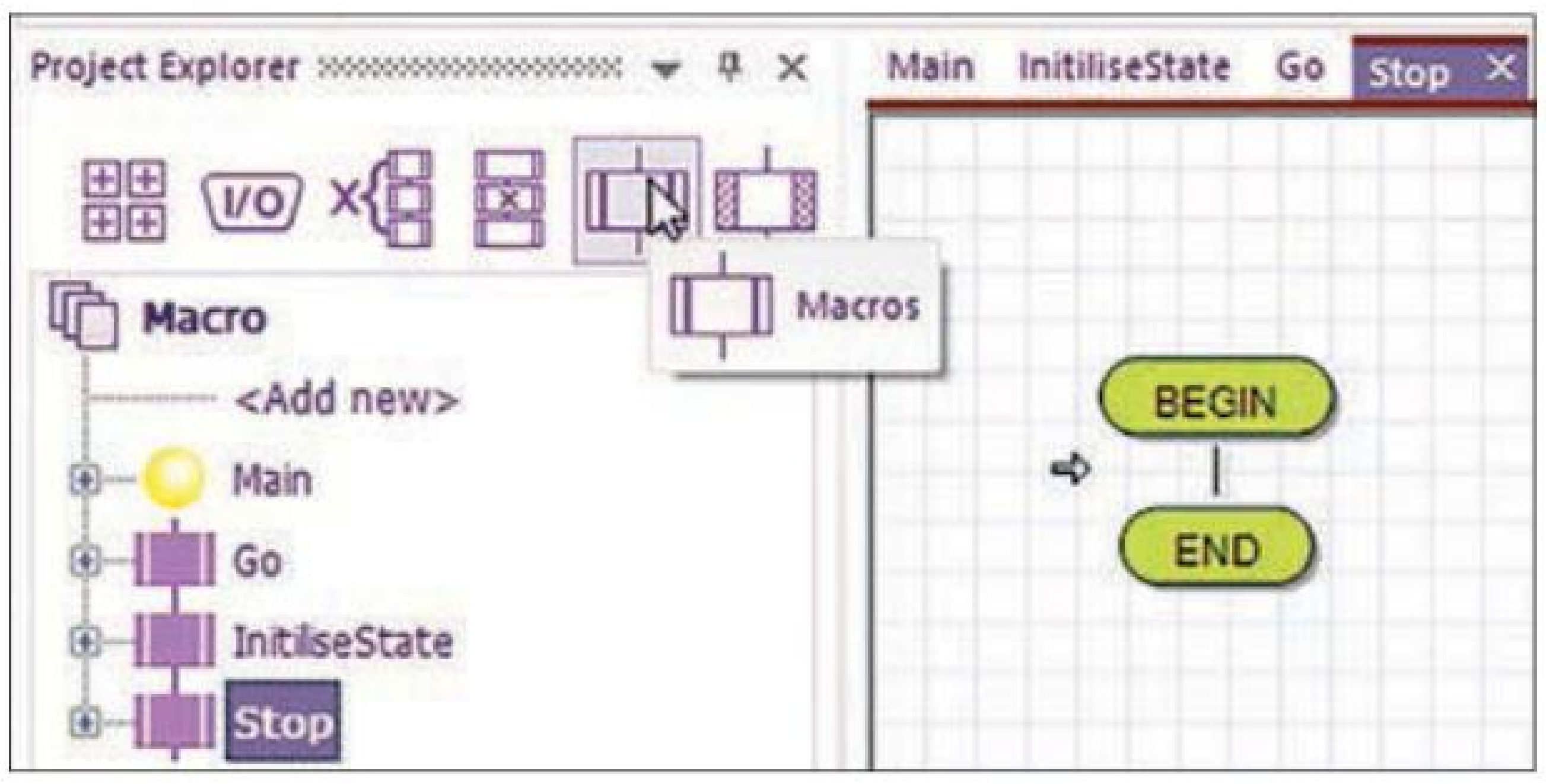


Fig.18. User macros.

To add User Macros, either use Project Explorer > Add New (or if you prefer, go via User Macros Ribbon > New Macro). In the resulting pop-up window add the Macro name (`InitialState`, `Go` or `Stop`) and click OK. At this point, your User Macros are just 'blank boxes'; later we will add their required functionality.

After the `InitialState`, `Go` and `Stop` User Macros have been created, select the Macros icon within Project Explorer and you will see the items shown in Fig.18. You will also see at the top of the working area, above the grid, four tabs, one for `Main` and three others for our new (blank) User Macros.

Variables

Programs almost always use elements called 'variables'. A variable is for holding values that can change. Its range (the values it can take) depend on the type of variable you choose (integer, Boolean, bytes and so on). The larger the range, the more memory the variable takes. Therefore, when deciding what type to use, it's important to choose carefully to minimise memory usage. In particular, make your choice based on what the maximum value will be.

In our traffic lights program, we only need to use one variable, which represents the next traffic lights sequence to be run (from a choice of two, red to green, green to red). I named this variable `Sequence`.

Let's check what kind of variables are available for assigning to `Sequence`. The easiest way to do this is to select the Global Variables Icon in the Project Explorer. (You can see this in Fig.19, where I have labelled all the icons you can select within Project Explorer – including Global Variable.) After selecting the Global Variables icon, select Add New, then Add New again in the resulting pop-up window.

Fig.20 shows the pop up window variable type options, long with the range of values each one offers.

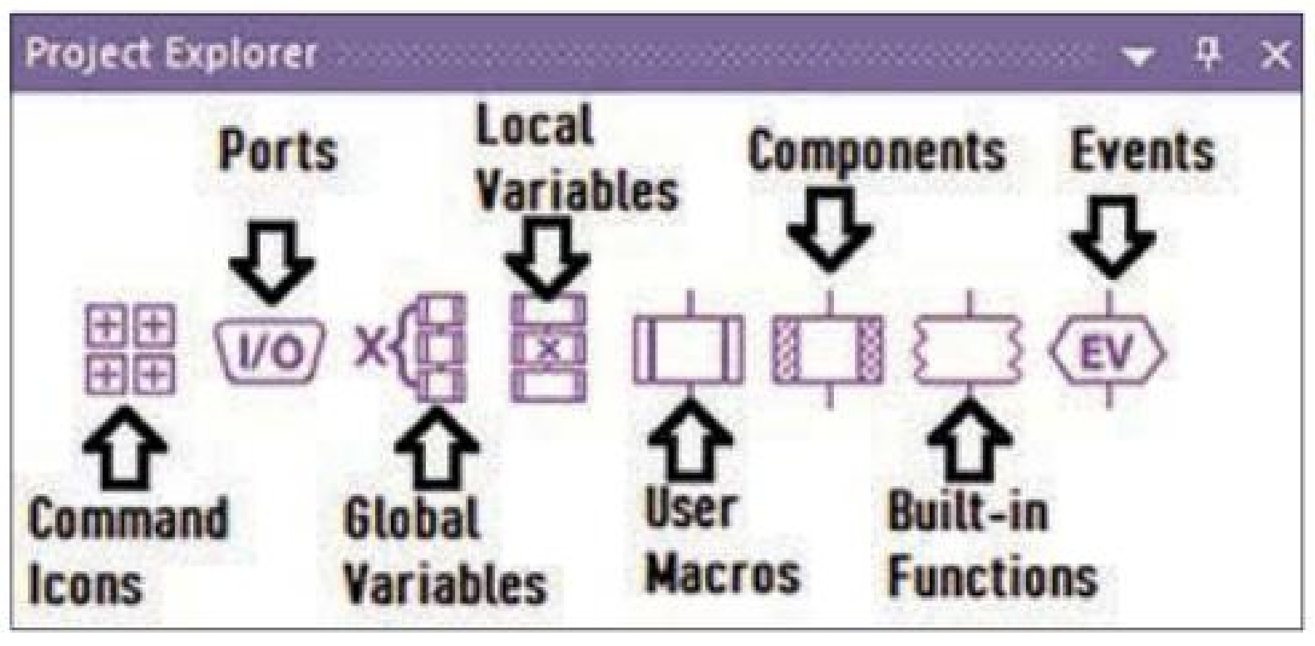
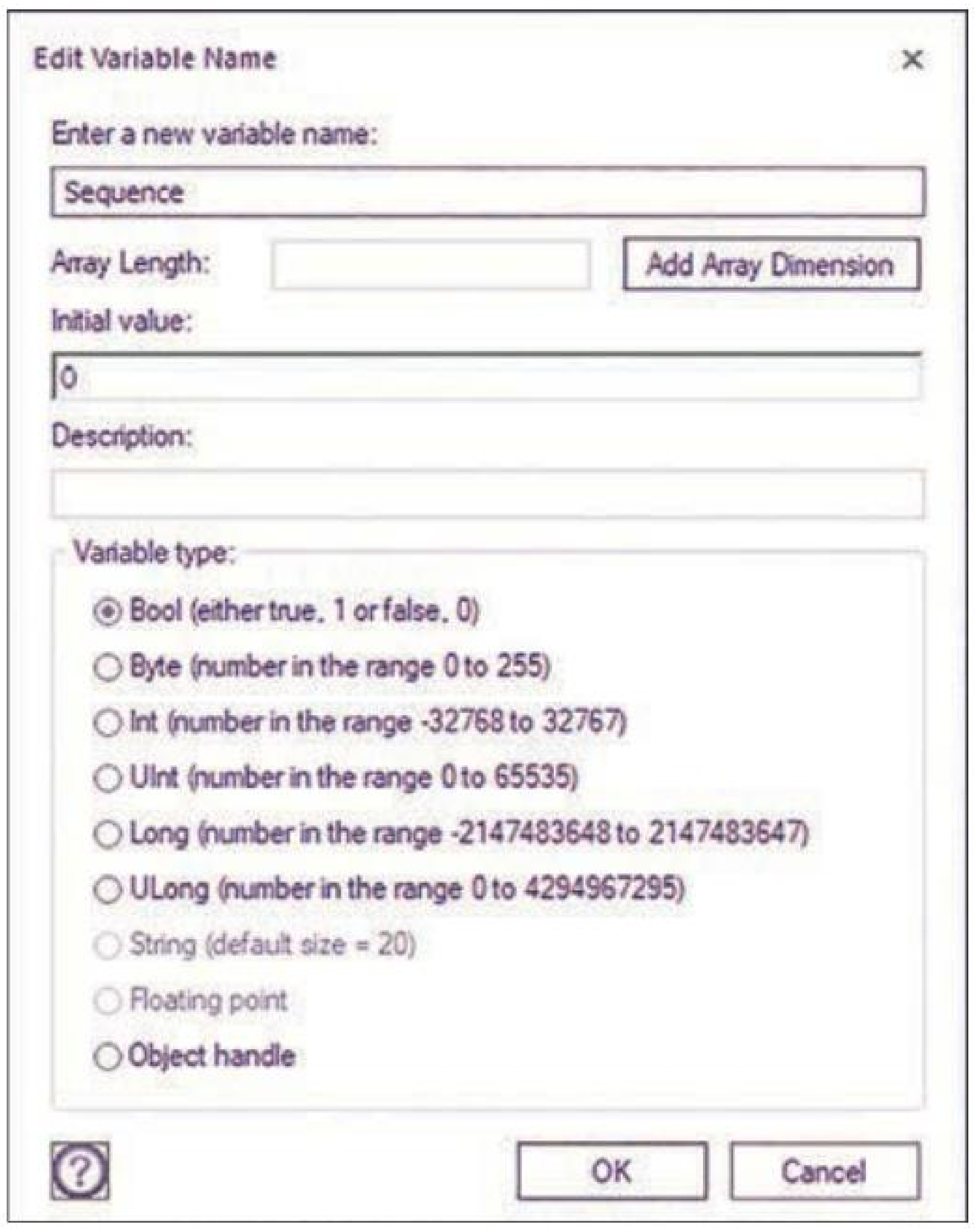


Fig.19. Use these guide to access Project Explorer Icons.



Edit Variable Name

Enter a new variable name: Sequence

Array Length:

Add Array Dimension

Initial value: 0

Description:

Variable type:

● Bool (either true, 1 or false, 0)
○ Byte (number in the range 0 to 255)
○ Int (number in the range -32768 to 32767)
○ UInt (number in the range 0 to 65535)
○ Long (number in the range -2147483648 to 2147483647)
○ ULong (number in the range 0 to 4294967295)
○ String (default size = 20)
○ Floating point
○ Object handle

OK Cancel

Fig.20. (left) Variable labelling and range choices.

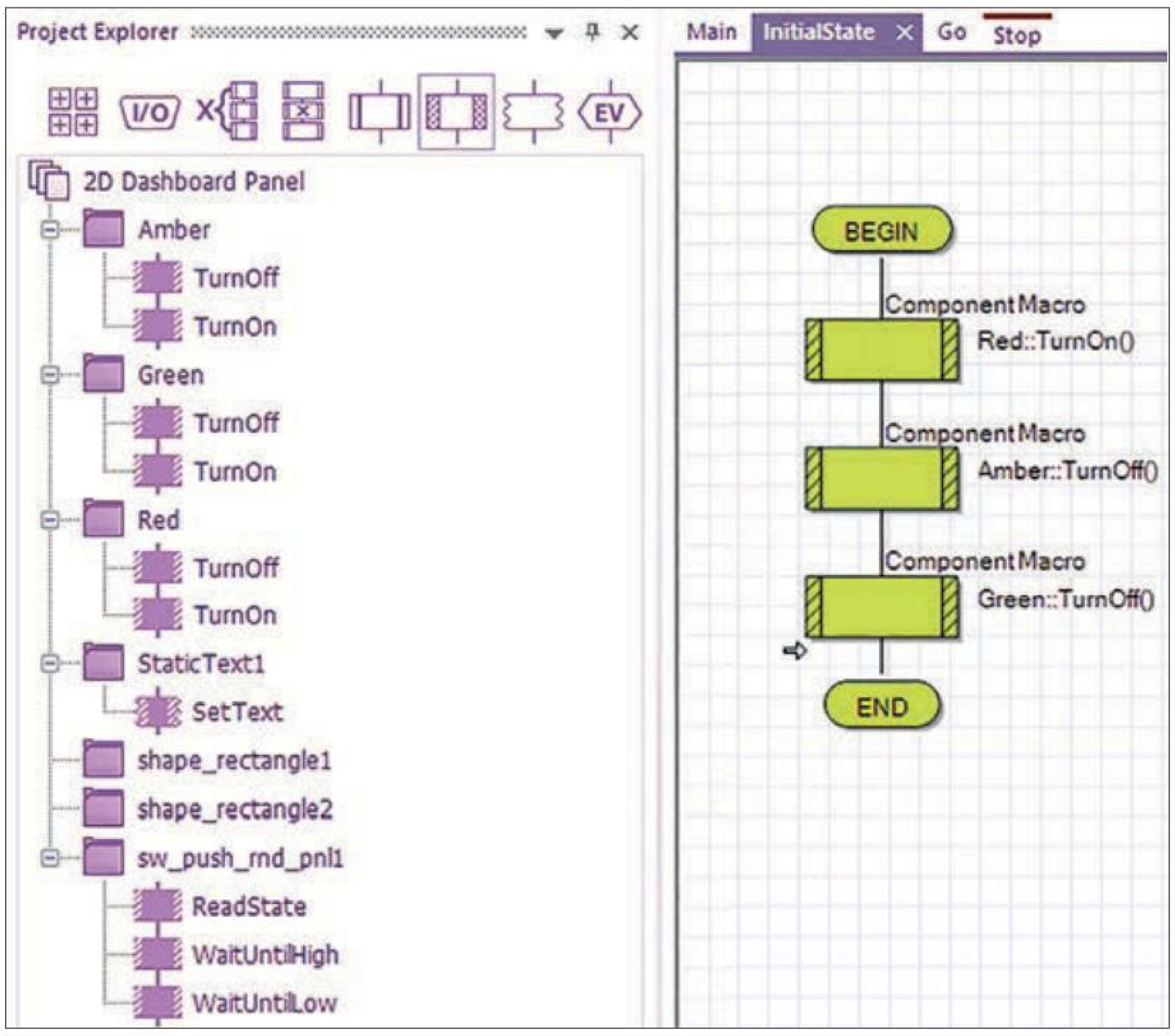

Project Explorer
2D Dashboard Panel
Amber
TurnOff
TurnOn
Green
TurnOff
TurnOn
Red
TurnOff
TurnOn
StaticText1
SetText
shape_rectangle1
shape_rectangle2
sw_push_rnd_pnl1
ReadState
WaitUntilHigh
WaitUntilLow
Main
InitialState
Go
Stop
BEGIN
Component Macro
Red::TurnOn()
Component Macro
Amber::TurnOff()
Component Macro
Green::TurnOff()
END

Fig.21. (right) InitialState User Macro Flowchart.

For our `Sequence` variable we are only interested in two, essentially binary values or states: 1 or 0, which will drive one of the two traffic lights sequences listed above. This means we choose variable type `Bool`, for Boolean (in the window of Fig.20). Name the variable `Sequence`. An initial value is required because otherwise our variable is undefined. Any variable that has not been set an initial value (uninitialised) will start with a random value. We want the `Sequence` value to start with 0, so place that in the Initial value box and then click OK.

Building the program

We are now in a position to build our program – the Main Macro, which is the program that is run on power up. Within the Main Macro we will 'call' User Macros to implement separate functions, such as initialising the traffic lights with a red light, running a sequence to turn the red light to green and then running a sequence to get back to the red light.

In each case, whether building the Main Macro or individual User Macros, we will do this by dragging icons to the working area. (Note that there are separate working areas for each User Macro, accessed via the top tabs. If you have a lot of tabs then it will be easier to select User Macros from the project explorer)

When the project is first run, the first thing we want to do is turn on the red light, we do this with the `InitialState` User Macro. So let's now build that User Macro, converting it from its current blank condition into one that can perform its intended function, which is simply to turn on the red LED.

Adding Component Macros

Let's start by adding a component. At the top of the working area click the `InitialState` tab. Next, recall that at the end of last month's article (final page) we created three LED components in Flowcode, conveniently called `Red`, `Amber` and `Green`. We now need to drag these three components, which are represented in the program as 'Component Macros', into the `InitialState` User Macro working area. Refer back to Fig.19, click on 'Components' and you will see the window in the left of Fig.21.

Note that our three LEDs each have two states – on or off. For our `InitialState` User Macro, we drag over the following: `Red-TurnOn`, `Amber-TurnOFF` and `Green-TurnOFF`.

Your flowchart should now look like the one on the right of Fig.21. (Don't forget to keep saving as you progress through the design build.)

Creating the Go User Macro

Now for the `Go` user macro. Click on the Go tab at the top of the window and add, in precisely this order, from top to bottom, the following:

```
Amber TurnOn
Delay 2 seconds
Red TurnOff
Amber TurnOff
Green TurnOn
```

The new element here is the Delay icon, which you will also find in the Command Icons section. Drag it to the working area, double click it and enter 2s. (See Fig.7 last month for further details.) The result should look like Fig.22.

Creating the Stop User Macro

For the `Stop` User Macro, strictly in the following order, add from top to bottom:

```
Green TurnOff
Amber TurnOn
Delay 2 seconds
Amber TurnOff
Red TurnOff
```

I might have made a deliberate mistake with the Stop user macro – let's find out when we debug!

That's it! We've created three user-friendly, intuitive User Macros with no hint of C, assembler or any other time-consuming, error-prone coding! Now all we need to do is bring them all together in the Main Macro. I hope you can see the power of using Macros to break down our program into easy-to-build chunks.

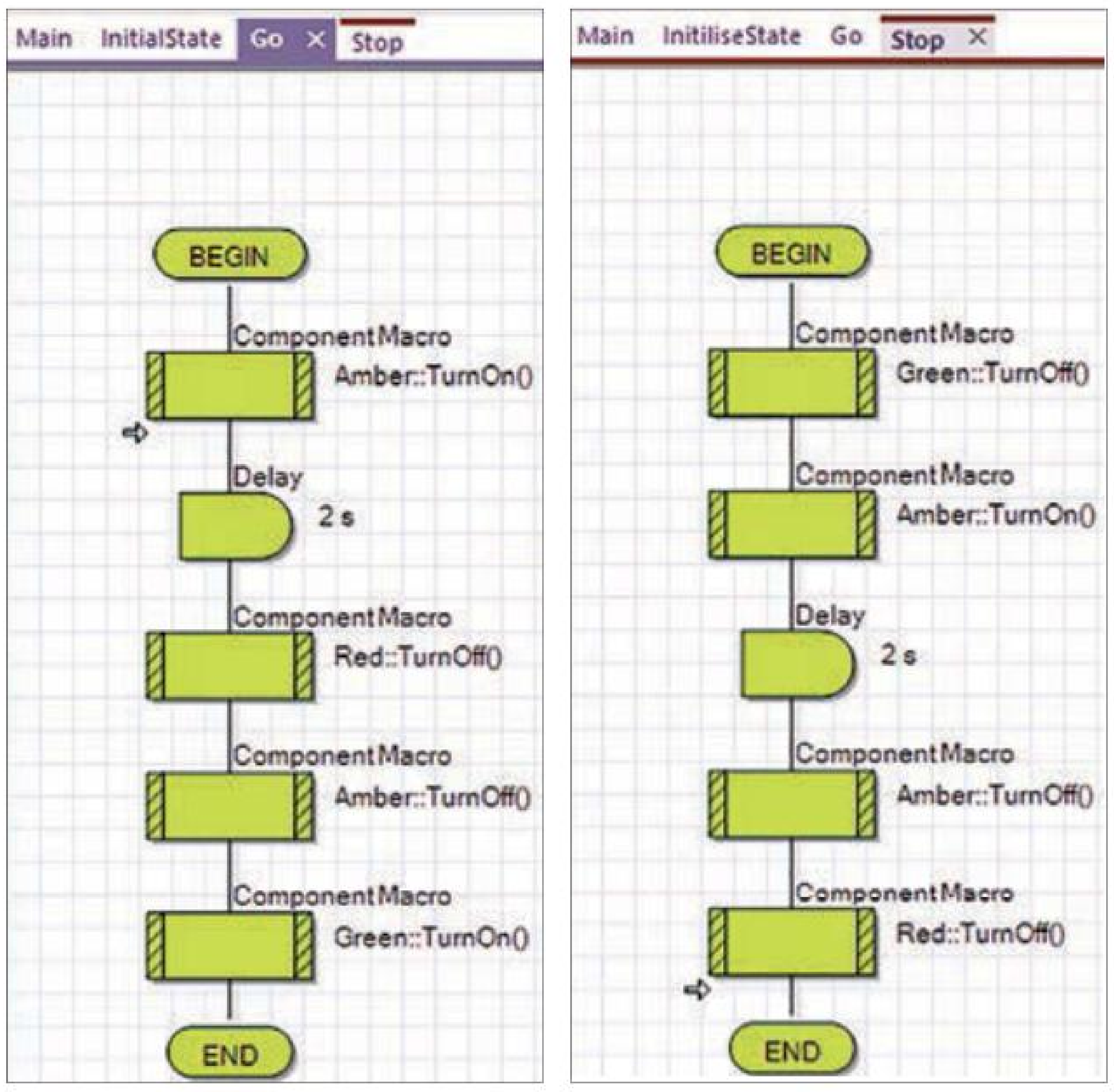

Main
InitialState
Go
Stop
BEGIN
Component Macro
Amber::TurnOn()
Delay
2 s
Component Macro
Red::TurnOff()
Component Macro
Amber::TurnOff()
Component Macro
Green::TurnOn()
END
Main
InitiliseState
Go
Stop
BEGIN
Component Macro
Green::TurnOff()
Component Macro
Amber::TurnOn()
Delay
2 s
Component Macro
Amber::TurnOff()
Component Macro
Red::TurnOff()
END

Fig.22. a) (left) Go user Macro flowchart; b) (right) Stop User Macro flowchart (... but is it corrrect?).

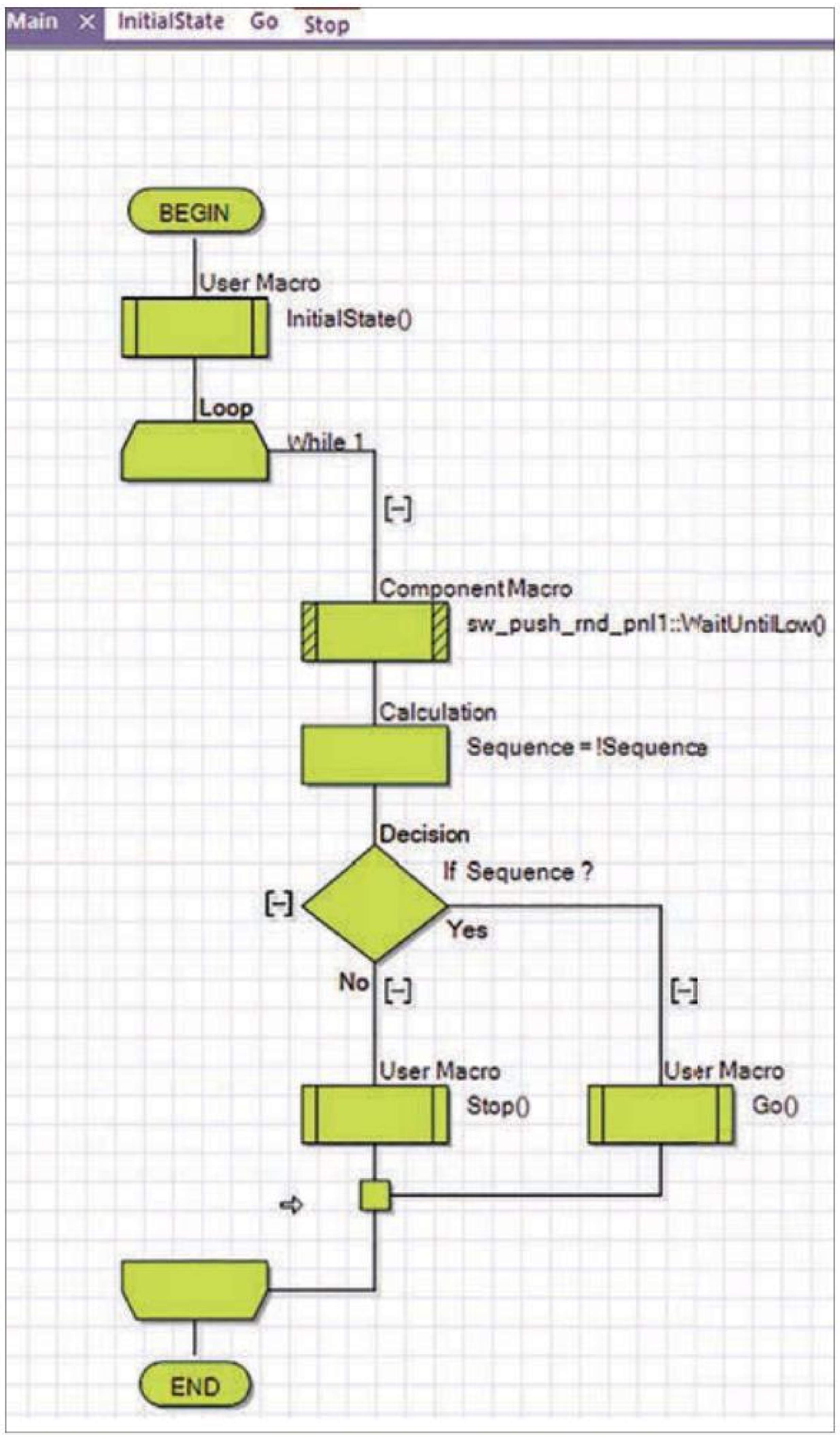


Fig.23. (right) Main Flowchart.

Creating the Main Macro

The final Macro we need to build is `Main`. As soon as out target powers up, we need it to call the `InitialState` user macro. So, we select User Macros from within Project Explorer, and drag the `InitialState` User Macro to `Main`.

Refer to Fig.23, which shows what we are building. To do this, drag the following, in the correct order, to below the `InitialState` user macro.

1. A `Loop` that keeps running forever (just like you did with the one-second LED flasher last month).

2. When we added the switch at the end of last month's article, I should have mentioned that it had the name `sw_push_rnd_pnl1`. Our switch has three options for our `Main` Macro. The one we want is `sw_push_rnd_pnl1 WaitUntilHigh` – drag it to the first position after the `Loop` icon.

The code will be paused at the switch until it's detected as being pressed. Next, we need to set some switch properties, double click the switch Component Macro, enter the following and then click OK:

```
Connection = $pin2
Polarity = Active low
Debounce = 50
Operation = Momentary
```

3. Next, we add a Calculation icon, double click it and assign it with `Sequence = !Sequence`. The `!` is a logical `NOT`, all it means is toggle the value of `Go`. If `Go` is `0` before the calculation, then it will be changed to `1` afterwards, and vice versa. We do this to set up the next block.

4. Now add a Decision branch (from the Command Icons – see Fig.4 from last month). This needs our `Sequence` variable in the `IF` input window. Double-click on the icon to open the IF window and make the following setting: immediately below `IF`, replace the `0` with `Sequence` and click OK.

The resulting decision branch has two options: `Yes` and `No`. If `Sequence` (as determined by our switch) is equal to `1`, then the `Yes` branch will be accessed, leading to User Macro `Go`. If `Sequence` is equal to `0`, then the `No` branch will be accessed, leading to User Macro `Stop`.

5. Finally, add the User Macro `Go` and the User Macro `Stop` in the two Decision branches.

Your Main Macro flowchart should now look like Fig.23. Hurrah! – That's it, you have now written the program, and I think you'll agree it involved not one line of 'written code'.

Running a simulation and debugging

We've created our program, so next we need to run it in simulation, and where necessary, debug it.

First, Select the Debug ribbon. (Switch on code profiling (Show Code Profiling) to highlight all the icons that are being accessed.)

To start the simulation in single-step mode, Press `F8`. You will notice a new window called Simulation debugger popping up. That will show you the Macro calls – ie, the `Main` or User Macros. It will also display the value of any variable that has been added.

Make sure you can see the traffic lights we drew last month on the 2D Dashboard panel. Press `F8` a few times, you will notice that the red light is on and the amber and green lights are off. So far, so good. If you carry on single-stepping it will stop at the switch as it waits for it to be pushed. (If it doesn't, you will need to check the switch properties have all been set correctly.)

Use the switch by selecting it and holding the left mouse button down, then single step until the red simulation outline is on the calculation icon, then the `Sequence` value will be `0`. Now release the mouse button, and start single-stepping again.

Observe what happens after the Calculation icon. The `Sequence` variable should change from `0` to `1`. Keep single-stepping, and the `Go` User Macro should be accessed, because the `Sequence` variable will be detected as `Yes`, and acted upon in the decision icon.

Keep single-stepping until you stop at the switch again. Are the lights correct on the traffic light? If not, check your `Go` user macro. Repeat the simulation again by single-stepping and clicking on the switch to change its state. This time the `Stop` user macro should be accessed.

Check if the correct lights are on. They shouldn't be! You will need to stop the simulation to change the choice of one of the LED component macro's states: in the `Stop` User Macro replace `Red TurnOff` with `Red TurnOn`.

Now, instead of single-stepping, press `F5` or select `Go` (`Run`). Left-click the switch for a few traffic lights sequences, which should now run correctly. When you are happy with the simulation, you can move on to start assembling the hardware.

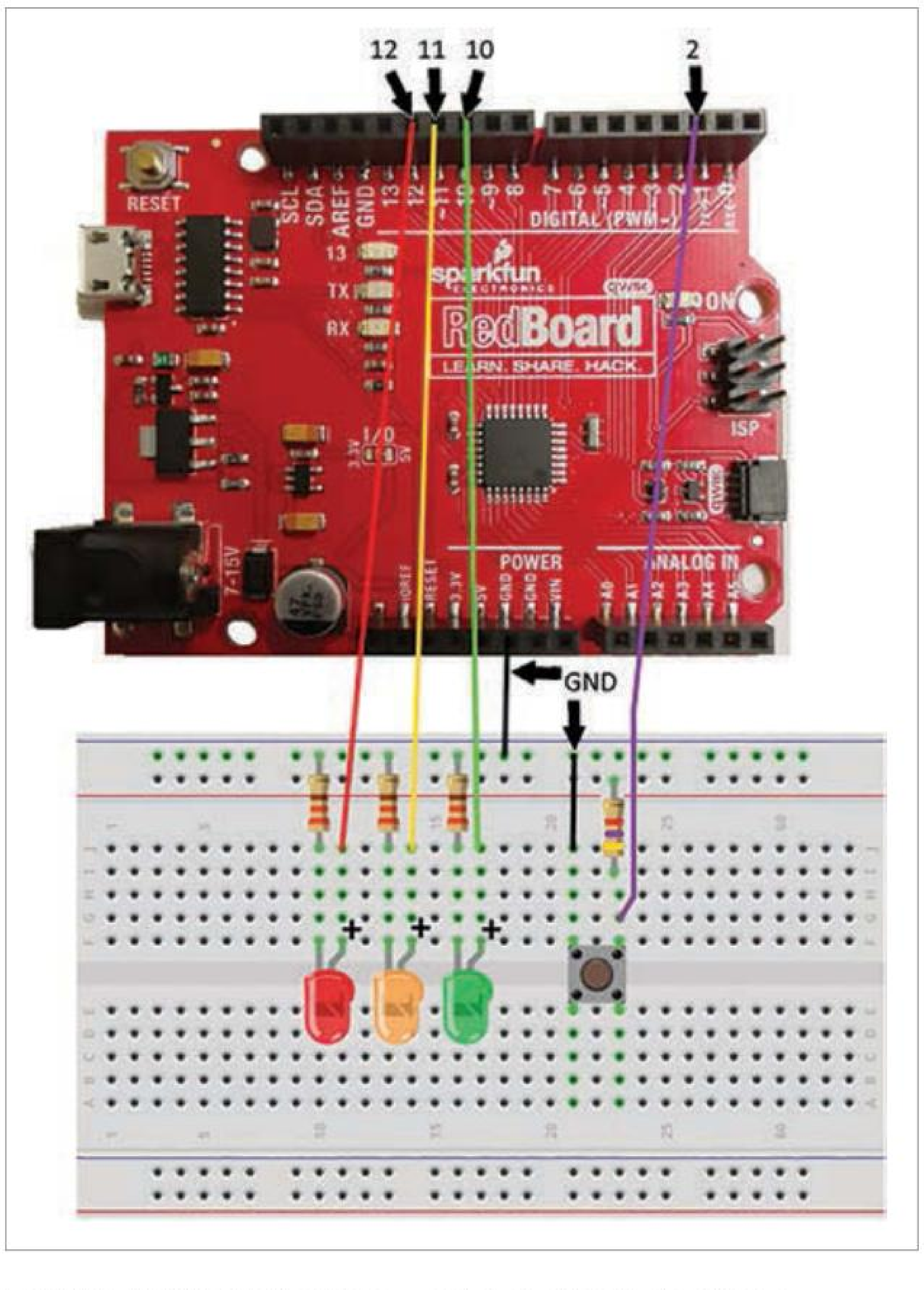


Fig.24. Wiring Diagram – add a circuit diagram.

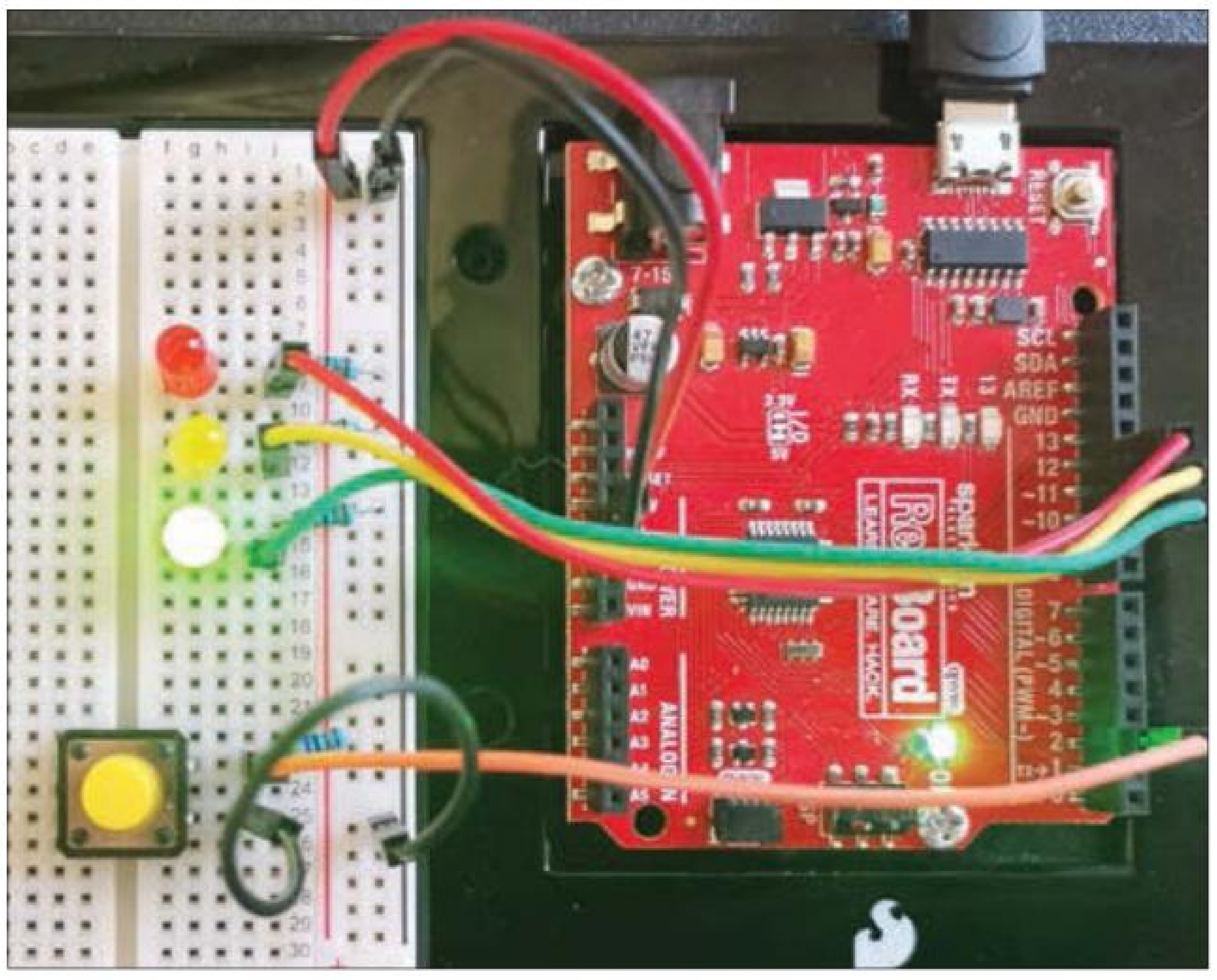


Fig.25. Traffic lights on green, after pressing switch.

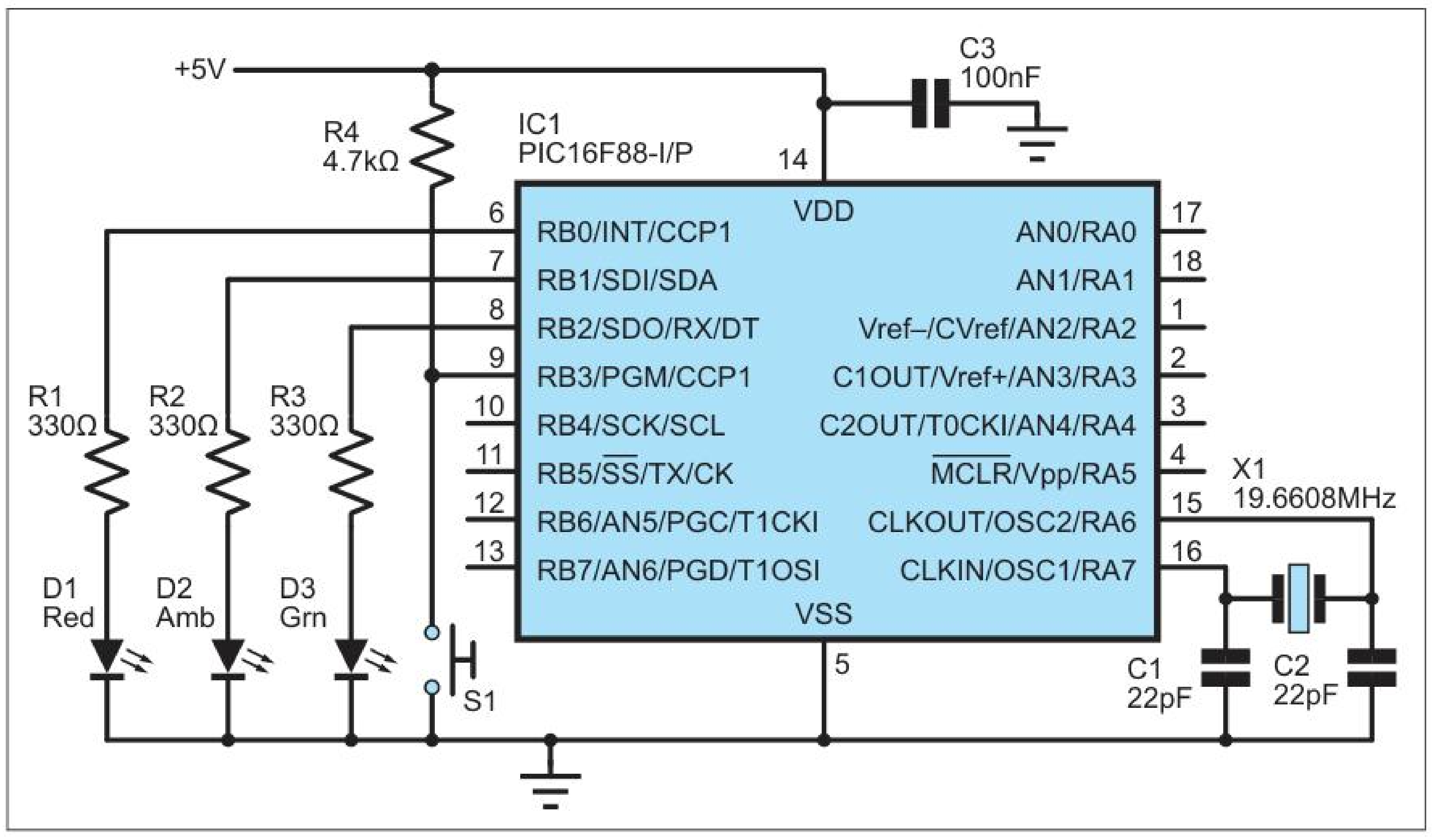


Fig.27. PIC16F88 circuit diagram to drive traffic lights.

Building the hardware

The hardware is straightforward – Fig.24 shows how to wire the RedBoard. If you have an Arduino Uno, the connections will be the same. Do note you need to include a pushbutton pull-up resistor, as show in Fig.24 and Fig.27 (Not listed in *Part 1*). When the wiring is finished you are ready to send the code to the target device, in exactly the same way the one-second flasher was programmed last month.

Now power it up. Once the hardware is running, just like the simulation, only the red LED should initially be lit.

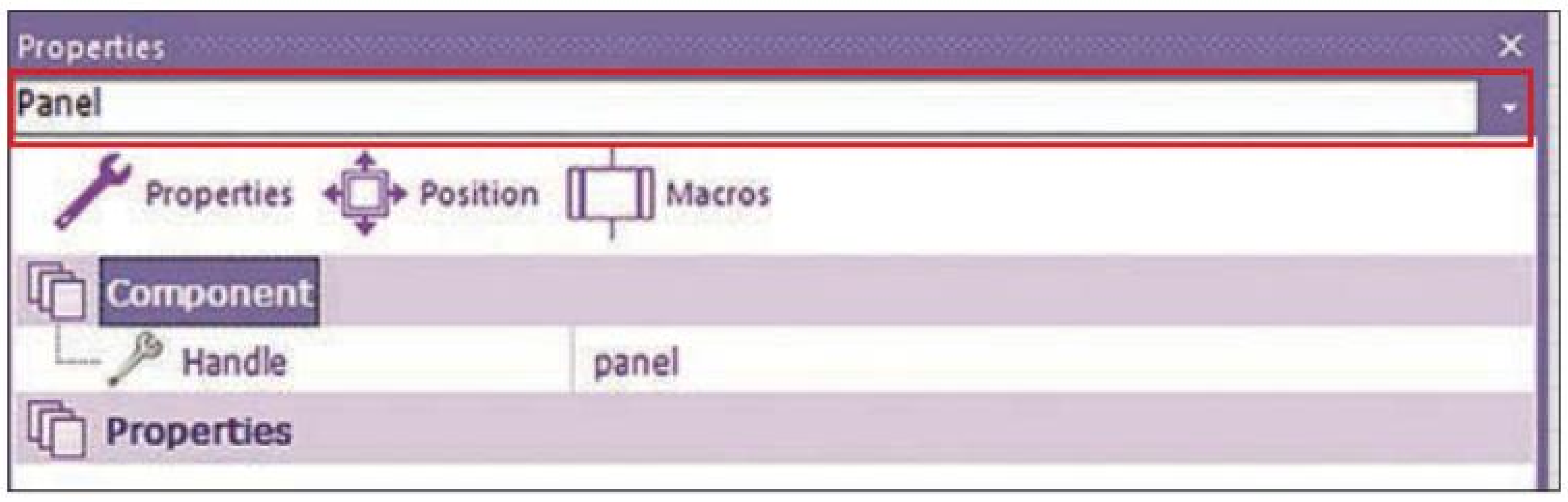
Then, when you press the button, the traffic lights sequence should change to green, in exactly the same way as real traffic light. This can be seen in Fig.25. Press the button again and it will cycle back to a red light.

Change the target to a PIC16F88

Last month, we only programmed a RedBoard, but this month we will demonstrate one of the most important strengths of Flowcode – code that is written for one target will work on other targets. So, this month we will program a PIC, the 16F88.

Changing the target device is easy. Just select the Build ribbon, Project Options. You will see a familiar window, as shown in Fig.1, last month, Enter 'F88' and choose '16F88'.

For a PIC, we need to choose our external crystal frequency – we will use a common value, 19.6608MHz (make sure you use a crystal with this frequency).



Properties

Panel

Properties | Position | Macros

Component

Handle	panel

Properties

Fig.28. Using View ribbon > Component properties.

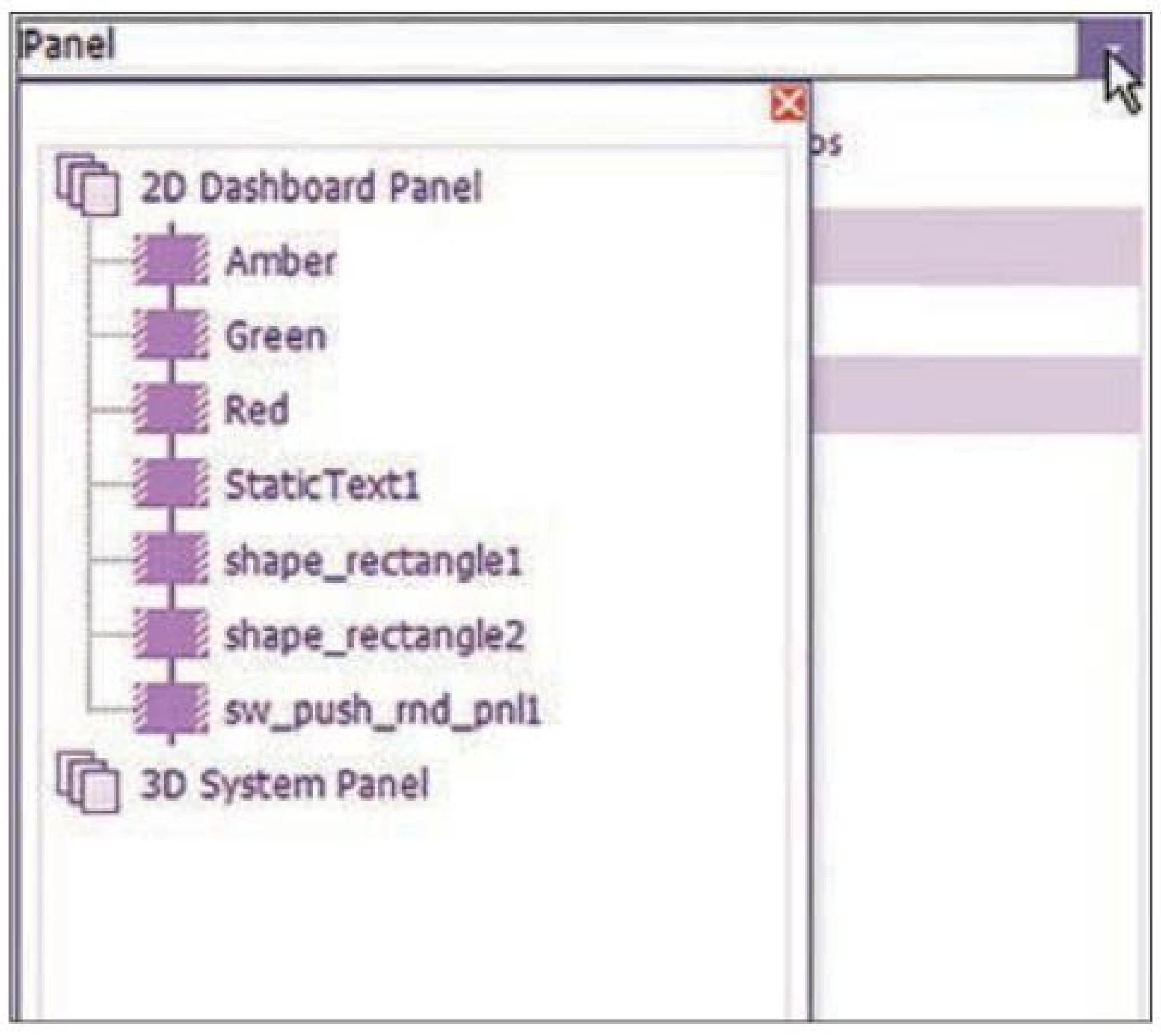
Now for the 'Configure' settings. We need make only one change in the settings. Change the `RA5/MCLR` Pin Function Select from `MCLR` to `RA5` – this means you don't need to use a pull-up resistor and you'll have an extra input-only pin. (Quick tip – if you don't see the above wording, then make sure the 'Show descriptive names' box is ticked.

When you have finished you should see what is shown in Fig.26, then select Modify.

Finally, we need to make changes to the properties of the LED and the switch Component Macros in Flowcode because the pin connections for a PIC are not the same as the RedBoard/Uno – see the circuit diagram in Fig.27. The new properties that need to be entered are as follows:

`Red`	`PORTB0`
`Amber`	`PORTB1`
`Green`	`PORTB2`
`Switch`	`PORTB3`

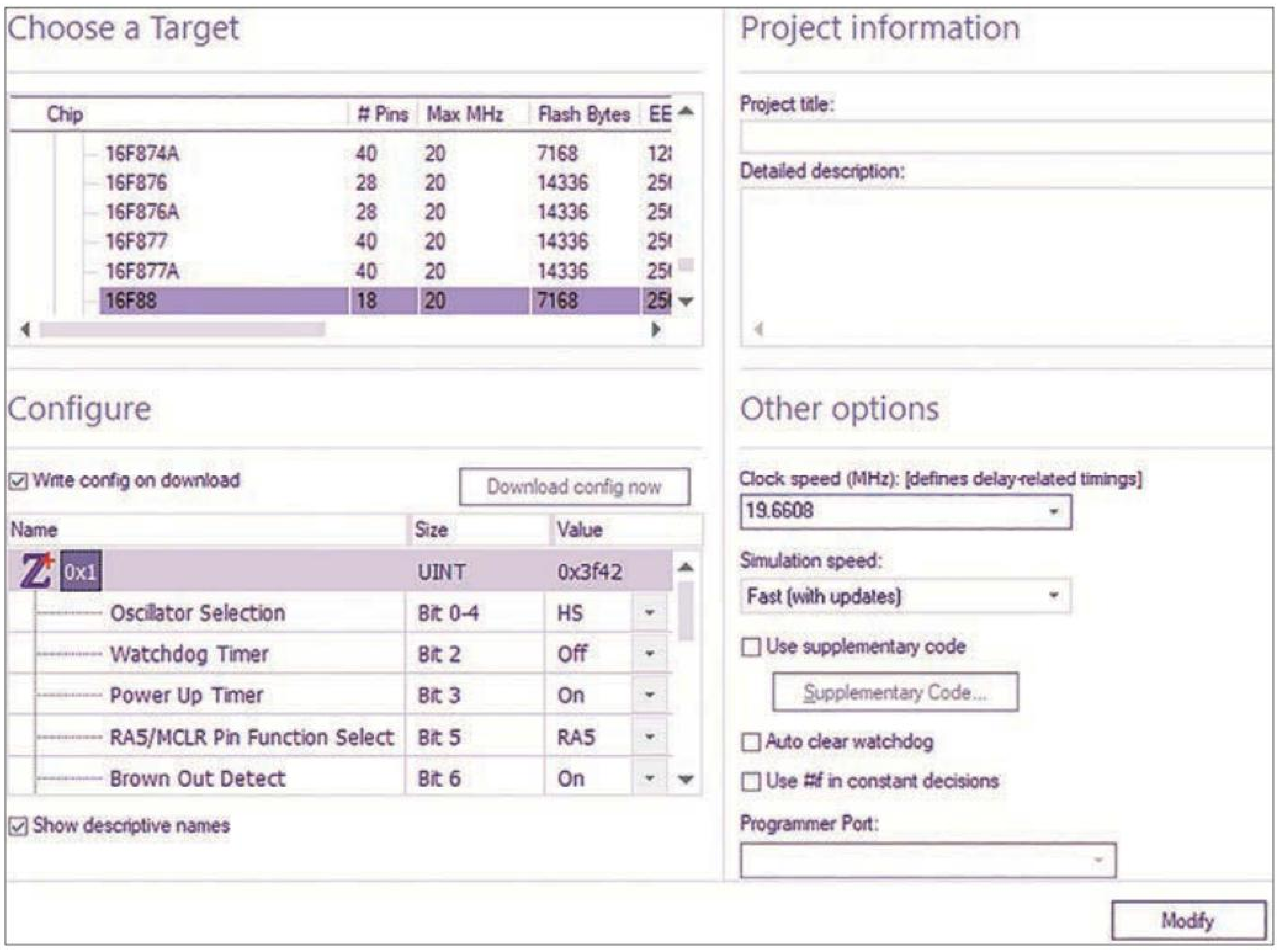
To do this, first go View ribbon > Component properties, and you will the window in Fig.28. Next, in this window click on the dropdown (here called 'Panel') and you will see the window in Fig.29. Click on the item you want to edit, for example 'Green', and you will see the Component Macro's Properties window (see example of this in Fig.9 last month).



Panel

- 2D Dashboard Panel
  - Amber
  - Green
  - Red
  - StaticText1
  - shape_rectangle1
  - shape_rectangle2
  - sw_push_rnd_pnl1
- 3D System Panel

Fig.29. The Component Macros.



Choose a Target

Chip	# Pins	Max MHz	Flash Bytes	EE
16F874A	40	20	7168	12
16F876	28	20	14336	25
16F876A	28	20	14336	25
16F877	40	20	14336	25
16F877A	40	20	14336	25
16F88	18	20	7168	25

Project information

Project title:

Detailed description:

Configure

☑ Write config on download

Download config now

Name	Size	Value
0x1	UINT	0x3f42
Oscillator Selection	Bit 0-4	HS
Watchdog Timer	Bit 2	Off
Power Up Timer	Bit 3	On
RA5/MCLR Pin Function Select	Bit 5	RA5
Brown Out Detect	Bit 6	On

☑ Show descriptive names

Other options

Clock speed (MHz): [defines delay-related timings] 19.6608

Simulation speed: Fast (with updates)

☐ Use supplementary code

Supplementary Code...

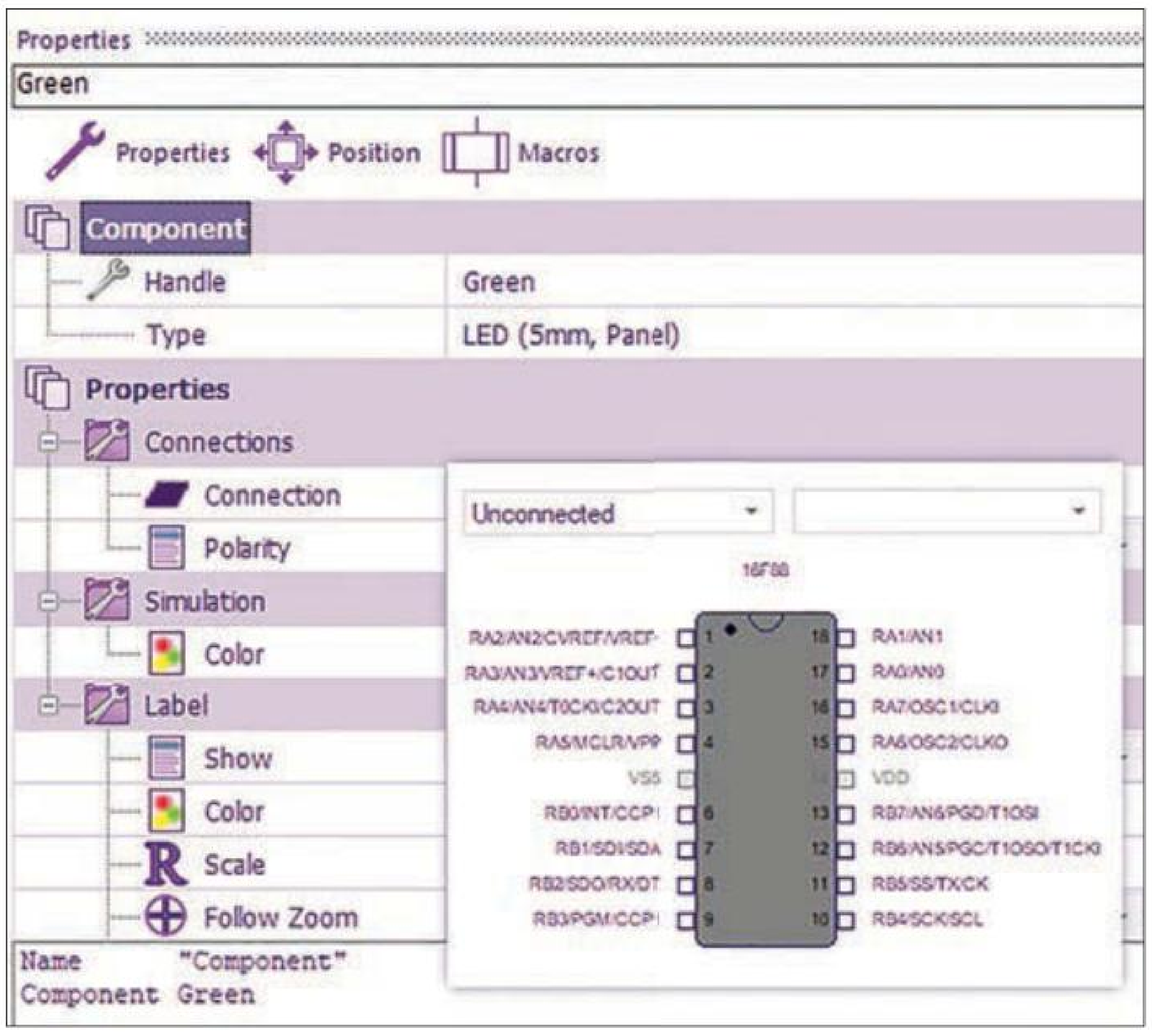
☐ Auto clear watchdog

☐ Use ## in constant decisions

Programmer Port:

Modify

Fig.26. PIC16F88 Configuration Settings.



Properties

Green

Handle: Green

Type: LED (5mm, Panel)

Connection: Unconnected

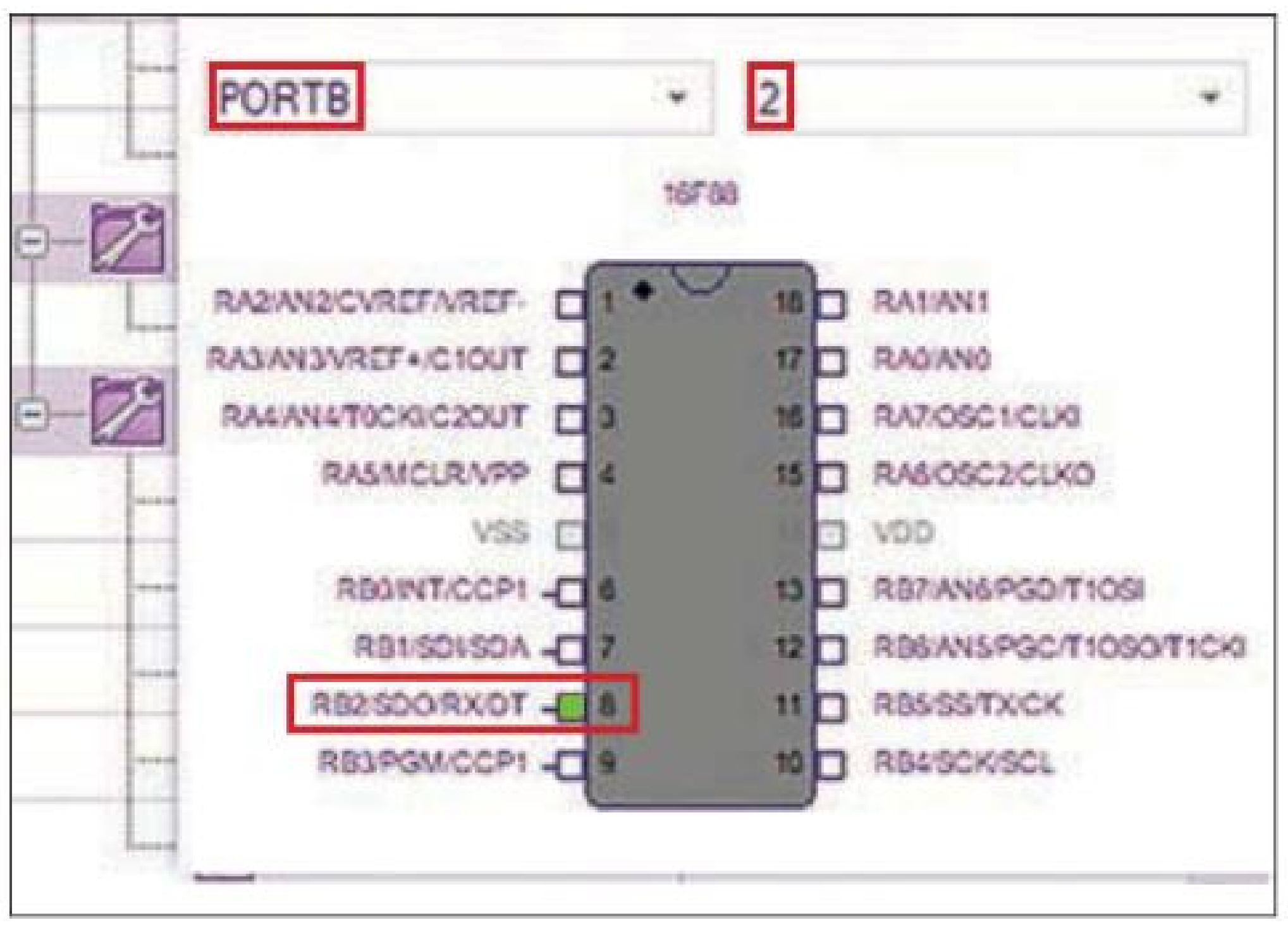
Name "Component"

Component Green

Properties
Position
Macros
Component
Properties
Connections
Connection
Polarity
Simulation
Color
Label
Show
Color
Scale
Follow Zoom
16F88
RA2/AN2/CVREF/VREF- 1
RA3/AN3/VREF+/C1OUT 2
RA4/AN4/T0CKI/C2OUT 3
RA5/MCLR/VPP 4
VSS
RB0/INT/CCP1 6
RB1/SDI/SDA 7
RB2/SDO/RX/DT 8
RB3/PGM/CCP1 9
18 RA1/AN1
17 RA0/AN0
16 RA7/OSC1/CLKI
15 RA6/OSC2/CLKO
VDD
13 RB7/AN6/PGD/T1OSI
12 RB6/AN5/PGC/T1OSO/T1CKI
11 RB5/SS/TX/CK
10 RB4/SCK/SCL

Fig.30. Assigning the PIC16F88 connections.

Now we can edit the Green LED's connection. Click on the choice box to the right of 'Connection'. (It may still have the RedBoard's connection and say something like, '$pin10'.) Once you've clicked, you will see a pop-up, as shown in Fig.30. Notice the two dropdowns: top left and top right. Click on the 'Unconnected' dropdown (top left) and it will give you a choice of PORTA or PORTB. We can use either, but we'll opt for PORTB because our crystal is connected to PORTA. Next, click on the righthand dropdown box and choose a PORTB connection – for Green, PORTB2. Now repeat the procedure for Amber, Red and the switch.



PORTB

2

16F88
RA2/AN2/CVREF/VREF- 1
RA3/AN3/VREF+/C1OUT 2
RA4/AN4/T0CKI/C2OUT 3
RA5/MCLR/VPP 4
VSS
RB0/INT/CCP1 6
RB1/SDI/SDA 7
RB2/SDO/RX/DT 8
RB3/PGM/CCP1 9
18 RA1/AN1
17 RA0/AN0
16 RA7/OSC1/CLKI
15 RA6/OSC2/CLKO
VDD
13 RB7/AN6/PGD/T1OSI
12 RB6/AN5/PGC/T1OSO/T1CKI
11 RB5/SS/TX/CK
10 RB4/SCK/SCL

Fig.31. The Green LED assigned to the PIC16F88's pin 8, PORTB2.

Notice that each time you select a connection the relevant PIC pin is highlighted in green on the IC profile diagram, as shown in Fig.31. Note the pin, so that you connect up the hardware correctly. And that's it, you have assigned each of the four Component Macros to a PIC pin.

It's worth reinforcing a really important point here. The RedBoard is a fully assembled microcontroller board based around an ATmega328 microcontroller IC. The PIC16F88 is a single IC from a completely different family of microcontrollers, but with just a few minor changes to our Component Macros we have ported our program from one to the other. No code rewrite needed. That flexibility is a key advantage of Flowcode.

Next, in Flowcode, select the Build Ribbon and Compile to Hex. A hex file will be generated in your project folder for you to load via your programmer of choice.

We are now ready to program the PIC16F88 using, for example, a PICkit 4 with MPLAB – see Mike Hibbett's excellent *PIC n' Mix* series, especially *Part 1-5* (July, September 2020 and January, March, May 2021). Do make sure your board is powered by an external 5V supply before programming.

Now test your hardware with a power-up and using the push-button in the same way you tested the RedBoard circuit.

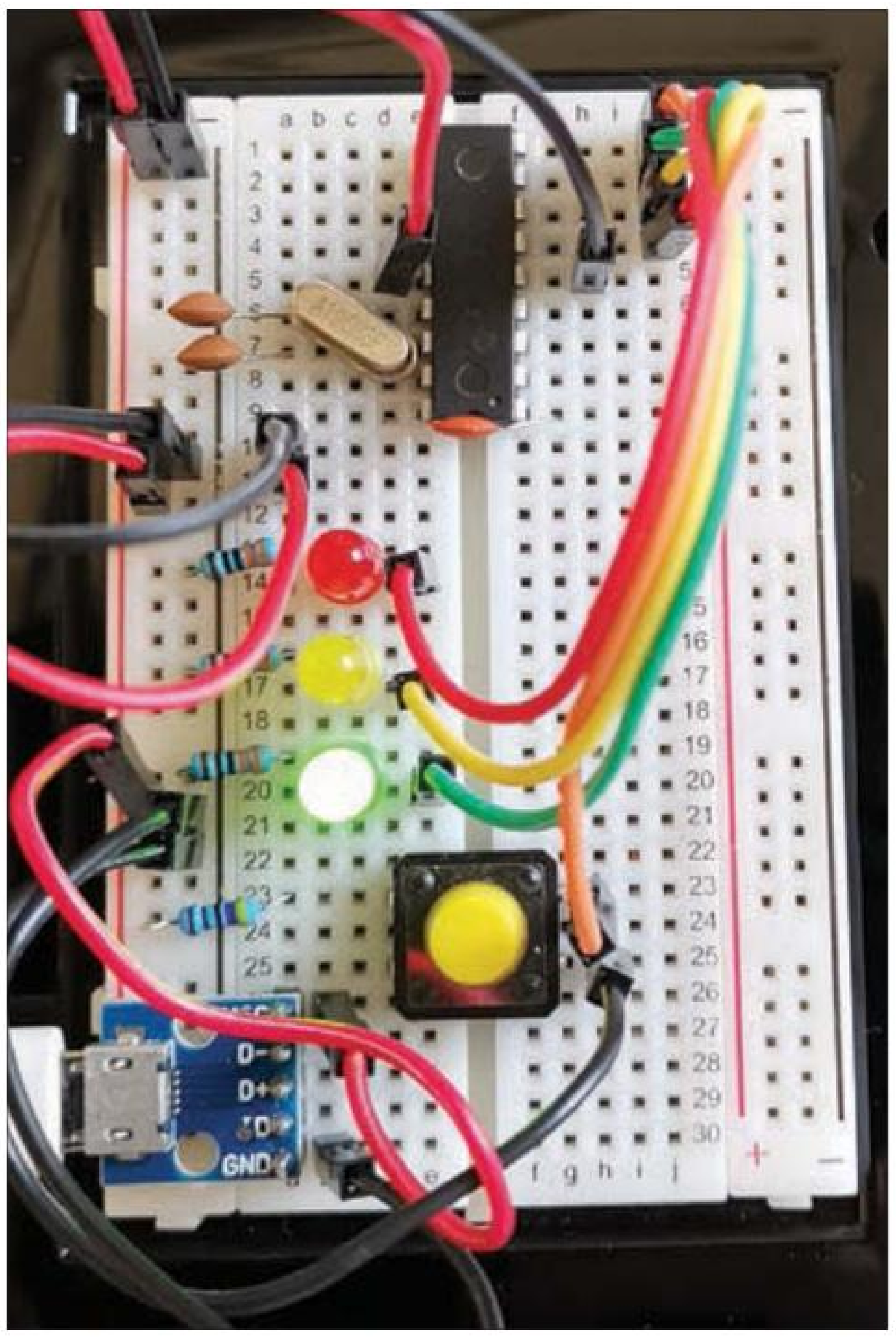


Fig.32. Traffic lights on green, after pressing switch.

Fig.32 shows my PIC circuit operating after programming.

We've covered a lot of ground, and the first time you use a powerful application like Flowcode there are a lot of ideas and processes to absorb, but I hope you can see that the fundamental nature of programming with Flowcode is quick, simple and intuitive – whatever your target.

Next in Flowcode

Our next Flowcode challenge will be to connect a 16x2 LCD to a RedBoard and PIC16F88. We will explain how to format and send characters to display a variety of messages.

Martin Whitlock is an Applications Engineer at Matrix TSL – the company behind Flowcode.

Try Flowcode for *free!*

We hope you found this article interesting. If you'd like to try Flowcode for free then just go to **https://flowcode.co.uk/download/** and download the code. You'll get a 30-day free trial of the full version – but that's not all. Even after the 30 days are up your copy of Flowcode will continue to work, but at a reduced level with a limit on the size of program you can run and access to a more basic set of parts. However, for beginners it is still an ideal platform with which you can build and run programs, on for example, the Arduino Uno/RedBoard or a PIC 16F88 that we are using in this project.

Only when you are sure that you want to use Flowcode do you need to buy *inexpensive* access to say a Raspberry Pi or Bluetooth module. (See: **https://flowcode.co.uk/buy/process/** for all the modules available and what they contain.) What's more, as soon as you buy *any* module, the restrictions on the size of your code are removed.

If you get stuck with anything relating to Flowcode use (installation, software, creating flowcharts, compiling to hardware, hardware not working… then there are forums set up to help you: **https://www.flowcode.co.uk/forums**

PE discount!

One more thing – Flowcode is deliberately designed to be inexpensive, but *PE* readers can get a further 20% discount when they use the code **PE20** at checkout.

Max's Cool Beans

By Max the Magnificent

Flashing LEDs and drooling engineers – Part 24

What's that, you say? You want to hear a joke? Well, by some strange quirk of fate, I happened to hear something humorous on the radio on the way into work this morning. A young lad calls his mum to say: 'Remember how my teacher said I'd never be any good at poetry because I'm dyslexic? Well, I just made two candlestick holders and a vase, and they look great!' Stop groaning. You'll be telling this yourself before the day is out.

Wow! Wireless LEDs

Just when you think you've seen it all, along comes something new to surprise you. Recently, for example, a friend introduced me to the YouTube channel of someone who goes under the moniker of Atomic14, which is sort of clever because 14 is the atomic number of silicon. While I was meandering around this channel, as is my wont, I saw a video showing wireless LEDs: **https://bit.ly/3y5HUlX**

It has to be said that these little scamps, which are available for purchase from AliExpress (**https://bit.ly/3IyqpiN**), look rather spiffy (Fig.1.). I can easily imagine using these to create some little gadgets, gizmos, doodads, and doohickeys to entertain young kids – or even (especially) old kids like myself, now I come to think about it.

I'm a SMAD boy!

We covered a lot of ground in my previous column (*PE*, January 2022), including toroidal transformations, gas-heated soldering irons, 1970s oil-wheel disco lighting effects, the 4-axis joysticks we'll be using to control our servo-driven robot heads, and some interesting considerations with respect to connecting multiple potentiometers in parallel. So, you can only imagine my surprise to discover that most of the emails I received regarding that column asked, 'But what about *SMADs?*'

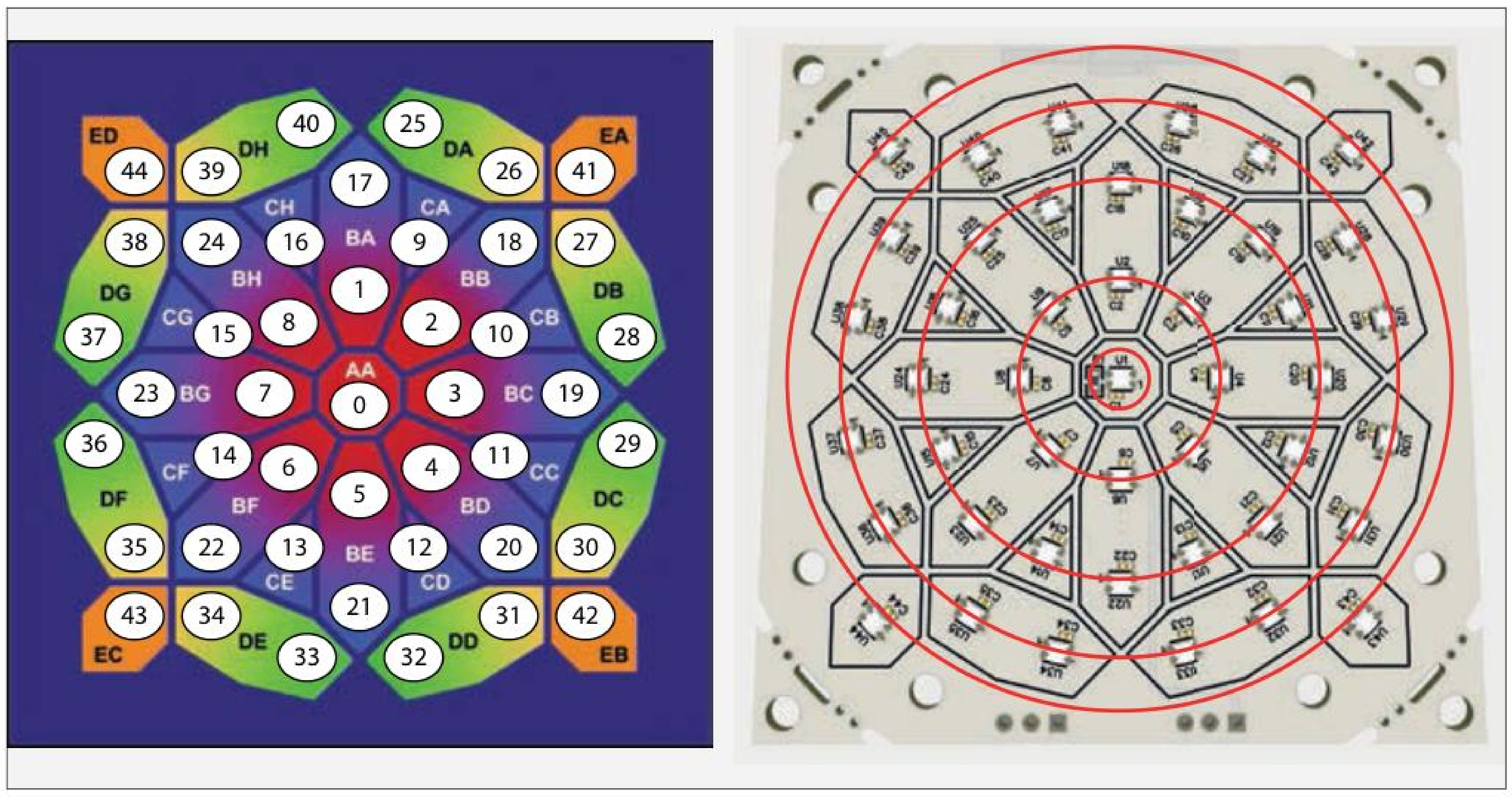


Fig.2. *SMAD* segment map (left) and board annotated with rings (right).

In fact, while I'm thinking about it, I'm even receiving emails from someone in Russia who used our *SMAD* schematic diagram to build a giant *SMAD* about 500mm square out of discrete WS2812 tricolour LEDs. He constructed this with his daughter, and they are constantly clamoring for more code examples.

Good grief! It's not like I've forgotten these bodacious beauties. It's just that I'm juggling too many balls and I can't juggle. Actually, that's not strictly true – I can juggle ten fine china plates, but only for a very short period of time; that is, the time it takes for them to go up in the air and come back down again (hey, it's a start).

In previous columns (*PE*, October and November 2021), we discussed a variety of simple windmill-type effects and an alien countdown timer effect. We presented these effects on two pseudo robot heads, each of which is equipped with two *SMADs* as eyes. The *SMADs* on one of these heads are augmented with 29-segment shells, while the other head flaunts 45-segment shells. Two columns ago (*PE*, December 2021), we had our heads talking to each other in Morse code. In fact, we even have a video of this taking place (**https://bit.ly/3oBwemz**). Now it's time to get a little more adventurous. As a starting point, we might decide to consider the LEDs on the *SMADs* as being presented in the form of five concentric circles or rings (Fig.2).

Do you recall the way in which we defined the 'patterns' in our windmill and countdown effects as a two-dimensional array called `EffectMap[][]`? Well, we're going to use the same technique here (Fig.3). The first dimension specifies the number of rows in the array. This corresponds to the number of patterns in our effect, which will be the five rings, in this case.

The second dimension specifies the LEDs that appear in each ring. Ring 0 in the center of the *SMAD* is formed from just one LED (shown as 0 in Fig.2). Ring 1 is composed of eight LEDs (shown as 1 through 8 in Fig.2). Rings 2 and 3 each contain 16 LEDs. And Ring 4 boasts only four LEDs. This means that the maximum number of LEDs in any of the rings is 16. The reason we set this dimension in the array to be 16 + 1 = 17 is because we use the first element in each row to define the

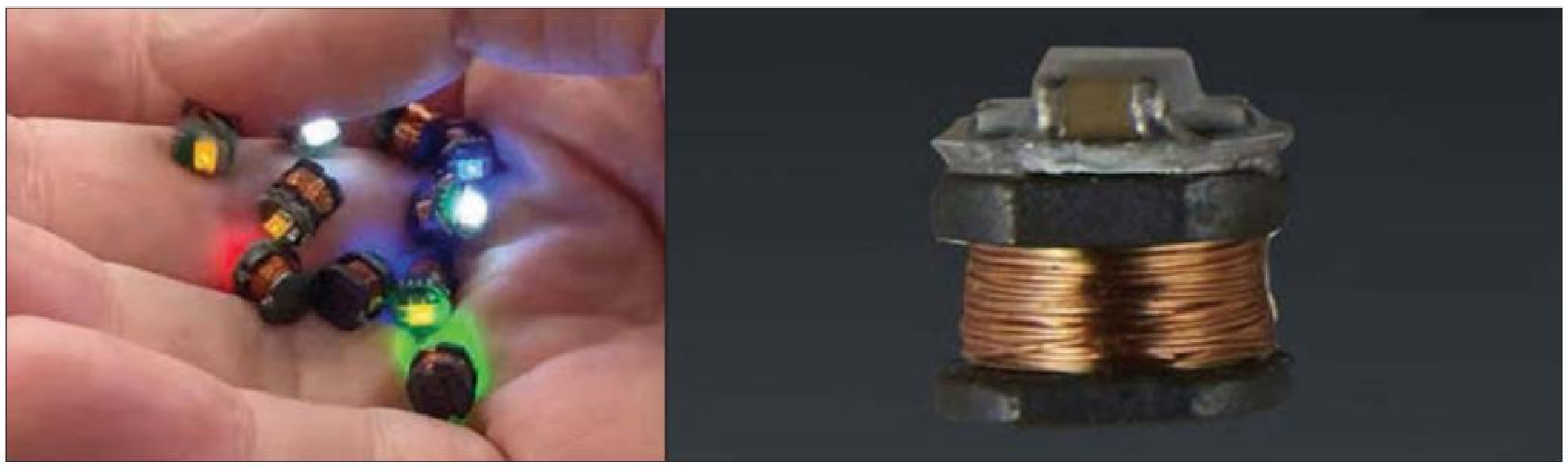


Fig.1. Wireless LEDs – I almost believe they were invented specifically for me...

```
#define  NUM_PATTERNS_IN_EFFECT  5
#define  MAX_LEDS_IN_EFFECT     16
const uint8_t EffectMap[NUM_PATTERNS_IN_EFFECT][MAX_LEDS_IN_EFFECT + 1] =
{
  { 1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0}, // Ring 0
  { 8,  1,  2,  3,  4,  5,  6,  7,  8,  0,  0,  0,  0,  0,  0,  0,  0}, // Ring 1
  {16,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24}, // Ring 2
  {16, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40}, // Ring 3
  { 4, 41, 42, 43, 44,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0}  // Ring 4
};
```

Fig.3. Specifying which LEDs appear in what rings.

```
#define  NUM_GROUPS        7
#define  MAX_LEDS_IN_GROUP 8
const uint8_t EffectMap[NUM_GROUPS][MAX_LEDS_IN_GROUP + 1] =
{
  {  1,  0,  0,  0,  0,  0,  0,  0,  0 }, // Group 0
  {  8,  1,  2,  3,  4,  5,  6,  7,  8 }, // Group 1
  {  8,  9, 10, 11, 12, 13, 14, 15, 16 }, // Group 2
  {  8, 17, 18, 19, 20, 21, 22, 23, 24 }, // Group 3
  {  8, 25, 27, 29, 31, 33, 35, 37, 39 }, // Group 4
  {  8, 26, 28, 30, 32, 34, 36, 38, 40 }, // Group 5
  {  4, 41, 42, 43, 44,  0,  0,  0,  0 }  // Group 6
};
```

Fig.4. Specifying which LEDs appear in what groups.

number of LEDs in that ring. Also, any '0' values are used only as placeholders (apart from the left-most '0' associated with Ring 0, of course).

My first test was simply to illuminate the rings with white light in a sequence starting with Ring 0 and radiating out to Ring 4. You can feast your orbs on the full sketch (program) by downloading the code (file **CB-Feb22-01.txt**) from the February 2022 page of the *PE* website: **https://bit.ly/3oouhbl**

I'm so random

For my next experiment, I decided to gather my LEDs into like-minded groups. There are seven of these groups numbered from 0 to 6. Group 0 corresponds to Ring 0 in our previous example and therefore contains just one LED. Similarly, Group 1 corresponds to Ring 1 in our previous example and therefore contains the 8 innermost LEDs in our eight radiating spokes.

In the case of Ring 2, we are going to divide this into two groups – Group 2 contains the eight LEDs in the eight little triangular segments, while Group 3 contains the eight outermost LEDs in our eight radiating spokes. Similarly, we are going to divide Ring 3 into two groups – Group 4 contains the eight 'anticlockwise' LEDs in our eight outer ring segments, while Group 5 contains the eight 'clockwise' LEDs in these segments. Last, but certainly not least, Group 6 contain the four LEDs corresponding to Ring 4 in our previous program (Fig.4).

What we are going to do is to randomly select a group and illuminate all the LEDs in that group with a randomly selected colour, and to continue to do this over and over again. If you wish, you can peruse and ponder this program (file **CB-Feb22-02.txt**).

For my third experiment, rather than change the colour of only one group at a time, I decided to change the colours of all the groups simultaneously (file **CB-Feb22-03.txt**). One point we should note is that, thus far, we've been displaying the same colour patterns on all the *SMADs*. It's really interesting to see how different the colours look when diffused by the 29-segment shells versus the stained-glass (some may say kaleidoscopic) effect offered by the 45-segment shells. Speaking of which, for the benefit of your cogitations and ruminations, I just took a video that shows all of the *SMAD*-related effects just discussed in this column: **https://bit.ly/3GwQb5f**

Gorgeous gammas

We introduced the concept of gamma correction in the context of my 12 x 12 ping-pong ball array in an earlier *Cool Beans column* (*PE*, January 2021). As you may recall, our human eyes have evolved to accommodate a huge dynamic range from moonlight to sunlight and – as part of this – they have a sort of built-in non-linearity. Gamma correction involves tweaking the values of the red, green, and blue channels in our LEDs to correct for this non-linearity.

In our case, the way we've been doing this in our programs is to use a gamma-correction cross-reference look-up table, the contents of which were kindly defined by the folks at Adafruit (Fig.5). We can use this table to remap the linear values we would *like to use* into their gamma-corrected counterparts that will provide us with the colours we *want to see*.

Personally, I thought our original gamma correction evaluations were rather successful, so I decided to apply this to our most recent *SMAD* experiment. In this case, for each robot head, I displayed the non-gamma-corrected colours on one *SMAD* and the gamma-corrected colours on the other *SMAD*.

One point that gave me pause for thought was that, in our original tests, we started off with our LEDs at full brightness insofar as the values required to represent the selected colour. When it comes to our *SMAD* experiments, however, I've been using a `ModifyBrightness()` function to lower the brightness of the LEDs so I don't wash out the sensor in my camera while taking videos.

This led me to wonder... should we modify the brightness and then perform the gamma correction (this is the way you'll see things in file **CB-Feb22-04.txt**), or should we do things the other way around. In the end, it didn't matter, because the results looked horrible both ways. I think one of our problems is that we are working with 8-bit values for our colour channels, which is too low a resolution for what we're trying to do. Also, we are using an overly simplistic approach (we should really be applying custom gamma correction functions for each channel based on the specific characteristics of our LEDs). I'm sure we could do better if we were desperate, but I wasn't desperate enough, so I decided to chalk this one up to experience and move on with my life.

Fabulous fades

If we journey once more back into the past to the early days of our 12 x 12 ping-pong ball array experiments, you may remember when we introduced some rather cool functions that allowed us to fade from one colour to another (*PE*, October 2020).

```
const uint8_t GammaXref[] =
{
      0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
      0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   1,   1,   1,   1,
      1,   1,   1,   1,   1,   1,   1,   1,   1,   2,   2,   2,   2,   2,   2,   2,
      2,   3,   3,   3,   3,   3,   3,   3,   4,   4,   4,   4,   4,   5,   5,   5,
      5,   6,   6,   6,   6,   7,   7,   7,   7,   8,   8,   8,   9,   9,   9,  10,
     10,  10,  11,  11,  11,  12,  12,  13,  13,  13,  14,  14,  15,  15,  16,  16,
     17,  17,  18,  18,  19,  19,  20,  20,  21,  21,  22,  22,  23,  24,  24,  25,
     25,  26,  27,  27,  28,  29,  29,  30,  31,  32,  32,  33,  34,  35,  35,  36,
     37,  38,  39,  39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  50,
     51,  52,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,  66,  67,  68,
     69,  70,  72,  73,  74,  75,  77,  78,  79,  81,  82,  83,  85,  86,  87,  89,
     90,  92,  93,  95,  96,  98,  99, 101, 102, 104, 105, 107, 109, 110, 112, 114,
    115, 117, 119, 120, 122, 124, 126, 127, 129, 131, 133, 135, 137, 138, 140, 142,
    144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 167, 169, 171, 173, 175,
    177, 180, 182, 184, 186, 189, 191, 193, 196, 198, 200, 203, 205, 208, 210, 213,
    215, 218, 220, 223, 225, 228, 231, 233, 236, 239, 241, 244, 247, 249, 252, 255
};
```

Fig.5. Gamma-correction cross-reference look-up table.

Fig.6. A CRT-based television displaying white noise ('static').

When I was first trying to wrap my brain around this, it actually took a bit of thought to decide how to perform a fade over a specific number of steps. Remember that, for each step, we first have to split each 24-bit colour into its three 8-bit red, green, and blue colour channels. For each of these channels we calculate the new value based on the start value, the end value, the total number of steps, and the current step. Finally, we gather the three 8-bit channels back into a 24-bit colour. Phew!

Happily, we can reuse tweaked versions of our previously created functions. For our fifth and final experiment in this column, I removed our ghastly gamma correction and added some fabulous fades (file **CB-Feb22-05.txt**). As you'll see in the video, we start by creating a palette of randomly selected colours, then we assign these colours to the various groups and display the groups. While keeping track of the original palette, we generate a new palette, assign these colours to the various groups, and then fade from the old colours to the new colours. And, of course, we keep on doing all of this over and over again.

Once again, if you watch the aforementioned video, I think you'll find it really interesting to see how different the colours look when presented via the 29-segment versus the 45-segment shells.

That's a lot of noise!

When most people hear the word 'noise,' their knee-jerk reaction is to think of a cacophony of sound, but noise comes in many forms. When I was a young lad circa the 1960s, our family's analogue television contained a bunch of vacuum tubes along with a cathode ray tube (CRT) for the display. At night, after the TV stations had stopped broadcasting, the screen filled with ever-changing random patterns of 'snow on black' flickering pixels (considering this was so dynamic, it's somewhat paradoxical that it was called 'static'). This visual representation of noise was complemented by an audible hiss (Fig.6).

Both of these realisations – the visual static and the audible hiss – are of a form known as 'white noise.' This is so-called because it includes all frequencies (more correctly, it has equal power in any frequency of a given bandwidth), which makes it analogous to white light, which is a mixture of all visible wavelengths of light.

In audio engineering, electronics, physics and many other fields, we use terms like 'white noise', 'pink noise', 'brown' (or 'Brownian') 'noise', 'blue noise', 'violet noise', 'grey noise' and even 'black noise'. These terms refer to the different power spectrums associated with the noise signals. For example, while an instance of white noise might be the sound of falling raindrops, an illustration of pink noise could be the 'chuff, chuff, chuff' sound of an old-fashioned steam engine.

You might think that noise is a bad thing to have in a system, but – if you know what you are doing – it can be useful on occasion. For example, when a continuous signal in the analogue domain is converted into a corresponding representation in the digital realm, it's confined to a number of discrete states called quanta. This results in something called quantisation error.

The term dither (or dithering) refers to intentionally overlaying the original signal with a small amount of noise, which serves to randomise the quantisation error. In the case of a computer, dithering can be used to create the illusion of greater colour depth in images on systems with a limited colour palette. Dithering can also be used to minimise visual artifacts like banding in images.

Another example that may be a little closer to home is when performing analogue-to-digital conversion. Suppose we have an 8-bit analogue-to-digital converter (ADC). Now suppose that we superimpose a noise signal on top of the analogue signal feeding the ADC, where the peak-to-peak amplitude of this noise signal is equivalent to one least-significant bit (LSB) of the digital output. If we now sample the signal at four times the original sampling rate and then apply a low-pass filter to the digital output (where the simplest low-pass filter would be to average the four samples), the result is to make it look like we have a 9-bit ADC. (This is a bit too complicated to take any further here – you'll just have to take my word for it!)

Actually, the origin of the idea to apply dither is rather interesting. In the early days of WWII, airplane bombers used mechanical computers (essentially, boxes filled with hundreds of shafts and gears and cogs) to perform navigation and bomb trajectory calculations. The strange thing was that these computers were found to perform their calculations more accurately when the airplane was in the air with its engines running then when it was on the ground with its engines turned off. Eventually, engineers realised that the vibration from the aircraft reduced the effects caused by 'stiction' (sticky moving parts), so, instead of moving in short jerks, the vibration caused them to move more smoothly. In order to replicate this effect on the ground, small vibrating motors were built into the computers. The vibration from these motors was called 'dither' from the Middle English verb 'didderen,' meaning 'to tremble.'

You gotta love lava lamps

The reason for my waffling on about noise is that it features in the answer to a poser that has long puzzled me. I'm sure you've seen a lava lamp at least once on your journey through life. These little beauties were invented by the British entrepreneur Edward Craven Walker in 1963. For quite some time, they were considered to be amazingly cool and – dare I say it (yes, I do) – groovy.

A lava lamp consists of a glass vessel containing a coloured waxy mixture at the bottom. The rest of the vessel contains a clear or translucent liquid. When an incandescent light is placed under the lamp, in addition to lighting it up, the heat causes balls of the coloured waxy mixture to decrease in density and randomly rise and undulate their way through the clear liquid.

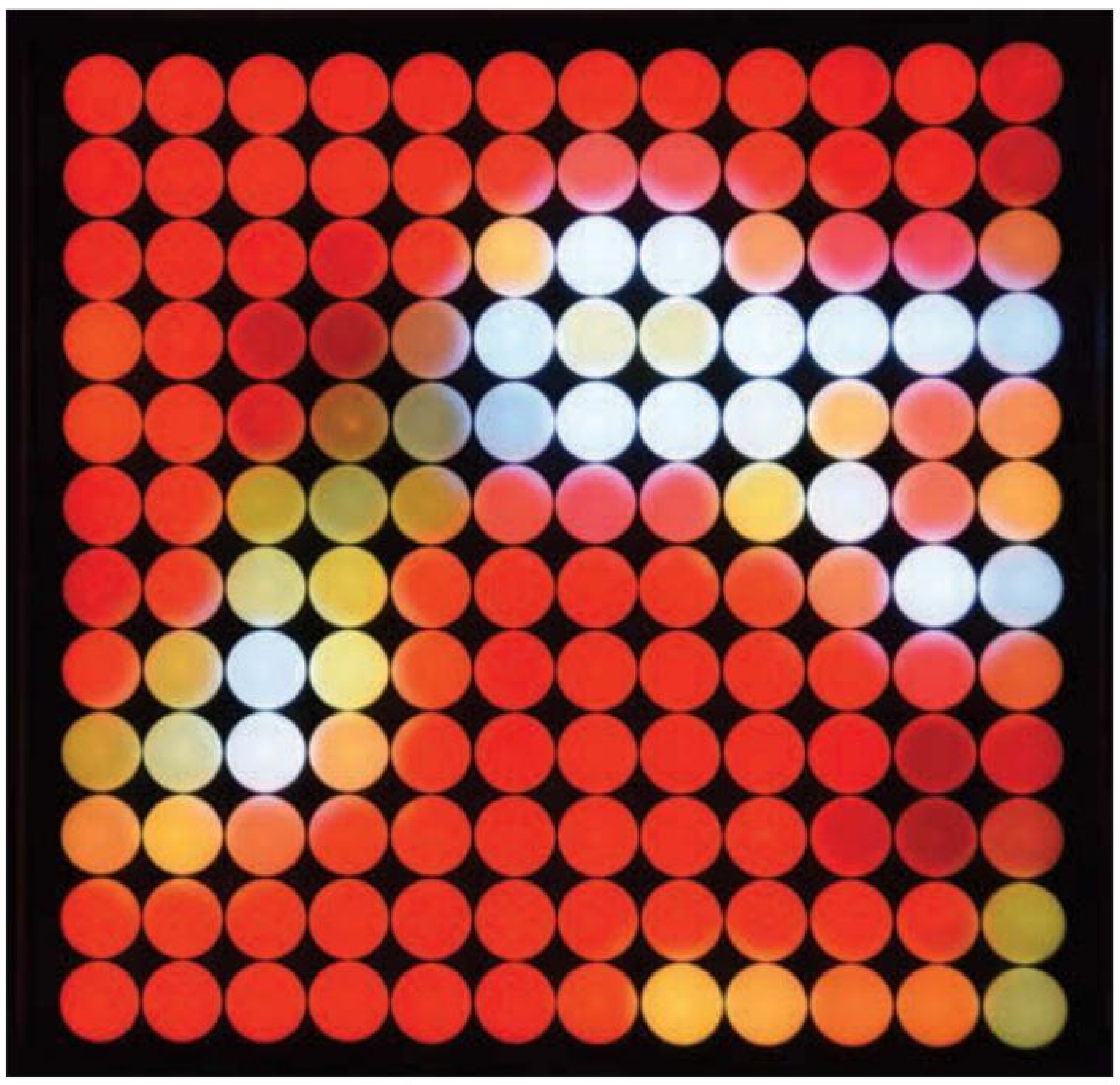


Fig.7. The FastLED NoisePlusPalette program meets my 12 x 12 ping-pong ball array – do watch the video!

As I mentioned in my previous column, on several occasions I've seen an effect presented on arrays of tricolour LEDs that is reminiscent of the oil wheels used in 1970s discos. This effect is also evocative of the undulations of the waxy balls in a lava lamp.

The FastLED library (**https://fastled.io/**) offers a great way to control tricolour LEDs like the NeoPixels I used to populate my 12 x 12 array of ping-pong balls. It turns out that there are a couple of example programs called 'Noise' and 'NoisePlusPalette' that are provided with this library. These programs use special noise functions that are based on something called 'Simplex noise', which was created by Ken Perlin in 2001, and which provides a method for constructing *n*-dimensional noise.

The great thing about Simplex noise is that it generates values that are pseudo-random overall but that are locally related at the same time. The easiest way to visualise the way this works is to imagine that you are looking at a giant oil lamp. If you were to take a series of photographs ten minutes apart, it would appear that there was no relation between the images. By comparison, if you took a snapshot every second, you could see how each image was related to the one that came before and the one that came after. That is, what you are seeing is both random and constrained at the same time.

The bottom line is that I tweaked the existing NoisePlusPalette program to drive my 12 x 12 ping-pong ball array (Fig.7). To be honest, in addition to restricting the program to using only its 'lava' palette, the main thing I had to change was the `XY()` function that accepts X and Y coordinates as inputs and returns the number of the NeoPixel in the chain (my NeoPixels are physically wired – daisy-chained – together in a serpentine pattern).

Want to build your own amazing array? All the details are in previous *Cool Beans* columns, starting in March 2020.

One thing I should note is that I originally tried to run this program on the 32-bit Seedunio XIAO microcontroller that I introduced in an earlier column (*PE*, July 2020), but... nothing happened. I thought I'd messed something up while performing my tweaks, and I was ready to pull out what little I have left of my hair when I thought to question my chum, Steve Manley. Steve did what I should have done, which was to have a quick Google. This revealed that there's a known problem with the XIAO and the FastLED library. A little later, Steve emailed me to say that he'd discovered a workaround (**https://bit.ly/3oJuvNn**), but he was too late. By this time, I'd already swapped out my XIAO for an Arduino Uno, after which everything worked just fine and dandy.

I'm really happy with the result, which you can see in a video I posted to my *Cool Beans* YouTube channel (**https://bit.ly/3ysgEhP**). Feel free to peruse and ponder the modified version of the program (file **CB-Feb22-06.txt**) but remember that I'm not the author of this code, just the tweaker.

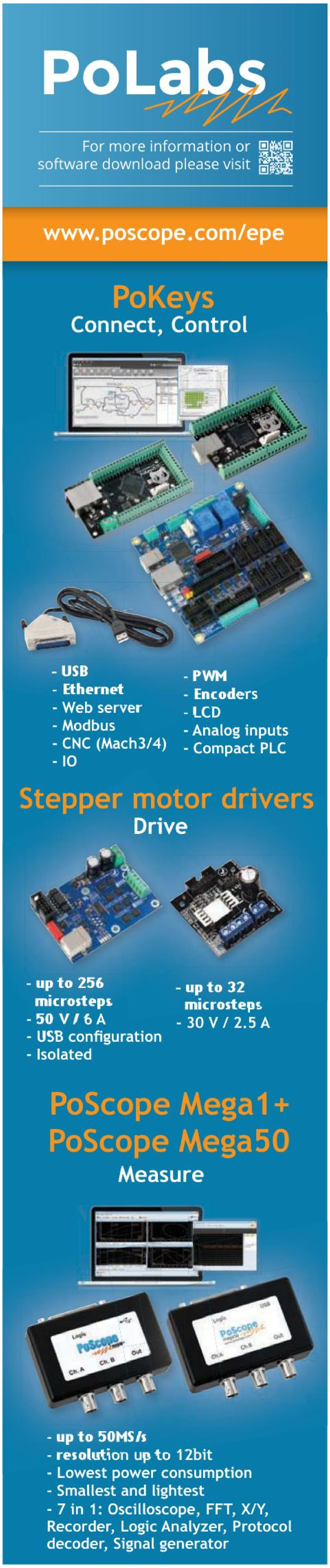
Don't lose your head

The definition of a cliffhanger is an adventure serial or melodrama that's presented in installments, each ending in suspense. When my mum was a little girl about 10 years of age, her older brother used to take her to the kids' show at the local cinema on Saturday mornings. The show was about two hours long and included a bunch of items, including a short thriller, like an episode of *Flash Gordon* or *Buck Rogers* (the old black-and-white versions from the late 1930s), which ended on a cliffhanger. My mom says she could barely control her excitement and it made her want to see the next episode all the more.

If only we could regain that sort of anticipation and suspense here. I know what we can do... in my previous column, I promised to show you the progress we've been making with regard to animating our robot heads. When I say 'we,' I mean me and my friend Steve Manley. And when I say 'all of the progress we've been making,' I mean all of the progress Steve has been making because he's storming ahead with something so super that you will squeal with delight when you see it. Sad to relate, however, this will have to wait until my next column (Cliffhanger – Tra-la!)

Cool bean Max Maxfield (Hawaiian shirt, on the right) is emperor of all he surveys at **CliveMaxfield.com** – the go-to site for the latest and greatest in technological geekdom.

Comments or questions? Email Max at: **max@CliveMaxfield.com**


PoLabs

For more information or software download please visit

www.poscope.com/epe

PoKeys
Connect, Control

- USB
- Ethernet
- Web server
- Modbus
- CNC (Mach3/4)
- IO
- PWM
- Encoders
- LCD
- Analog inputs
- Compact PLC

Stepper motor drivers
Drive

- up to 256 microsteps
- 50 V / 6 A
- USB configuration
- Isolated
- up to 32 microsteps
- 30 V / 2.5 A

PoScope Mega1+
PoScope Mega50
Measure

- up to 50MS/s
- resolution up to 12bit
- Lowest power consumption
- Smallest and lightest
- 7 in 1: Oscilloscope, FFT, X/Y, Recorder, Logic Analyzer, Protocol decoder, Signal generator

AUDIO OUT

By Jake Rothman

Analogue Vocoder – Part 4: Driver Amplifier

Last month, I introduced the balanced-input *Vocoder Driver Amplifier* which is used to drive the multiple filters used in the design. This month, we will complete the *Driver Amplifier*, looking at the design's output driver circuits and of course components and construction.

Transformer driver

For our *Vocoder*, the *Driver Amplifier's* output uses a balancing transformer, and hence it is a floating output which prevents ground loops. These can occasionally saturate on low frequency peaks, greatly increasing the current demand from the amplifier, which can result in clipping that may hang on for a few cycles. The additional current capability of this circuit reduces this horrid-sounding effect. Also, transformers give lower distortion with low-source impedance drive. Better still, is to make the output resistance negative to cancel out the transformer's primary DC resistance by using positive feedback. Fig.11 shows a circuit to do this, using a sensing resistor R_{SENSE}. This resistor value is dependent on the transformers primary DC resistance and the gain. Fig.12 shows the board wiring.

I used this circuit as an excuse to try a new transformer from Vigortronix, the VTX-102-3002, which has a very high saturation level of $37\mathrm{V}_{\mathrm{pk\text{-}pk}}$ at 30Hz into 600Ω. It has an upper frequency resonance at 88kHz which requires a Zobel network (C_z and R_z) to damp it. These transformers have a silicon-iron core, and this driver circuit reduces the distortion below that of some expensive nickel-core designs, while maintaining the high saturation.

High-impedance speaker driver

An additional use for this *Driver Amplifier* is as a low-power speaker/monitor driver. Most analogue musical products or systems, such as Eurorack synthesisers operate on ±15V rails and often a small speaker is required for monitoring. If a typical power amplifier chip, such as a TDA2030 is used with a standard 8Ω speaker, the output of 10W and the current consumption of over 1A is excessive. Using a high-impedance 50Ω speaker gives a power output of $1.2\mathrm{W}_{\mathrm{RMS}}$ with a current consumption of 50mA, which can easily be accommodated by normal op amp power rails. A similar power consumption situation arises when driving spring-lines reverberation units. For these low-power amplifier applications, heatsinks on the output transistor are necessary. For higher rail voltages, an 80Ω speaker should be used. A speaker driver combo is shown in Fig.13. Note that high-impedance speakers are available from the *AO Shop*, see components below.

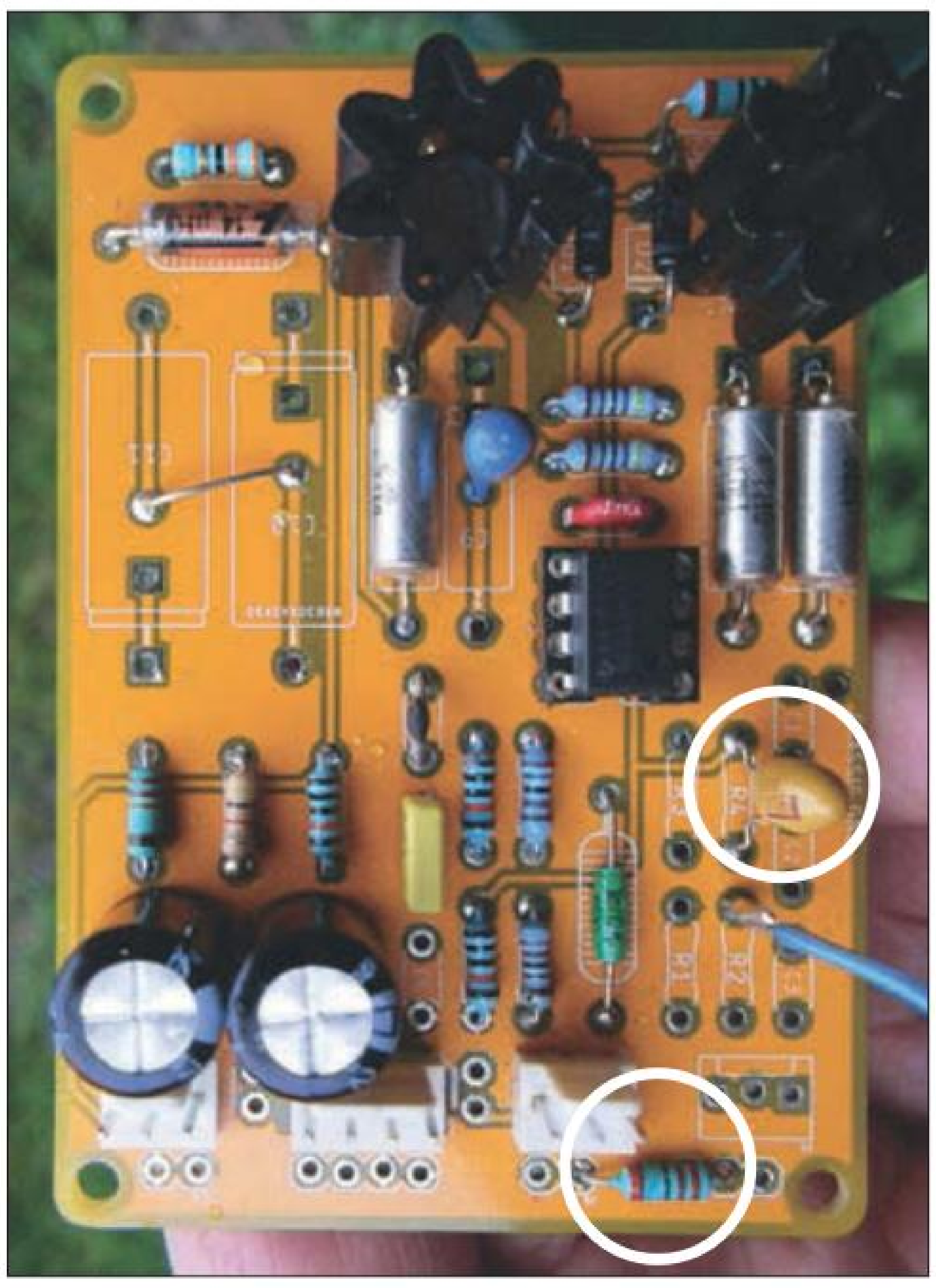


Fig.12. Assembled transformer output Driver Amplifier – this is basically a summing amplifier with a sensing resistor in the bottom right-hand corner and a tantalum bead coupling capacitor in the R4 position. The negative terminal goes towards the chip. The return from the transformer goes to the terminal pin inserted next to the capacitor positive in the hole for R2.

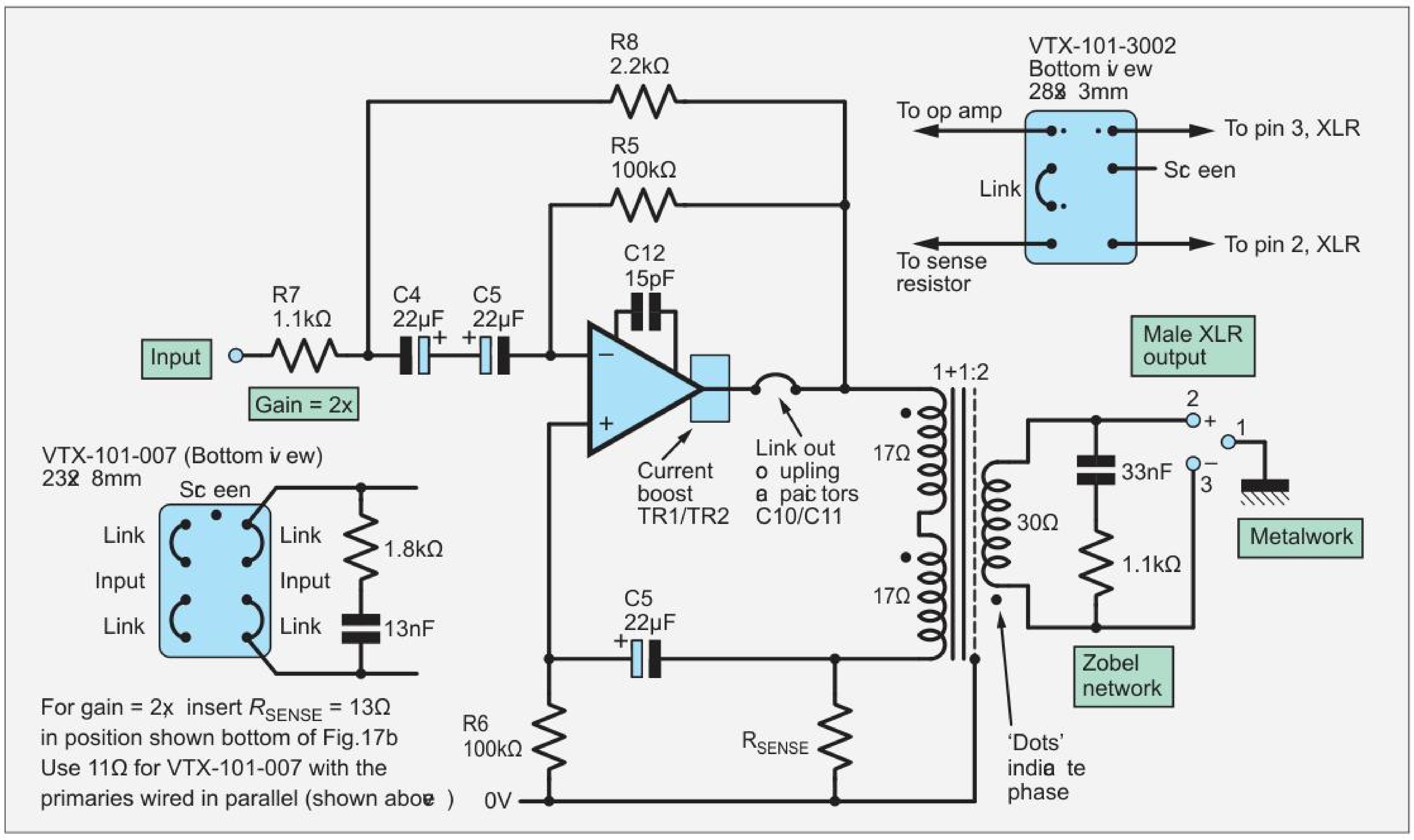

R8
2.2kΩ
R5
100kΩ
C12
15pF
R7
1.1kΩ
C4
22µF
C5
22µF
Input
Gain = 2x
VTX-101-007 (Bottom view)
23 x 28mm
Screen
Link
Input
Link
Link
Input
Link
1.8kΩ
13nF
Current
boost
TR1/TR2
Link out
coupling
capacitors
C10/C11
C5
22µF
R6
100kΩ
0V
RSENSE
VTX-101-3002
Bottom view
28 x 33mm
To op amp
Link
Screen
To pin 3, XLR
To sense
resistor
To pin 2, XLR
Male XLR
output
1+1:2
17Ω
17Ω
30Ω
33nF
1.1kΩ
2
1
3
Metalwork
Zobel
network
'Dots'
indicate
phase
For gain = 2x insert RSENSE = 13Ω
in position shown bottom of Fig.17b
Use 11Ω for VTX-101-007 with the
primaries wired in parallel (shown above)

Fig.11. Transformer driver circuit with negative resistance output to minimise distortion. Note the amplifier circuit is inverting and correct phase is achieved by flipping the secondary output wires to the XLR socket. If the sense resistor R_{SENSE} is too large the circuit will oscillate at a very low frequency – best optimised by experiment.

Bridge-drive output

Another useful, *non-Vocoder* application of this design is to use

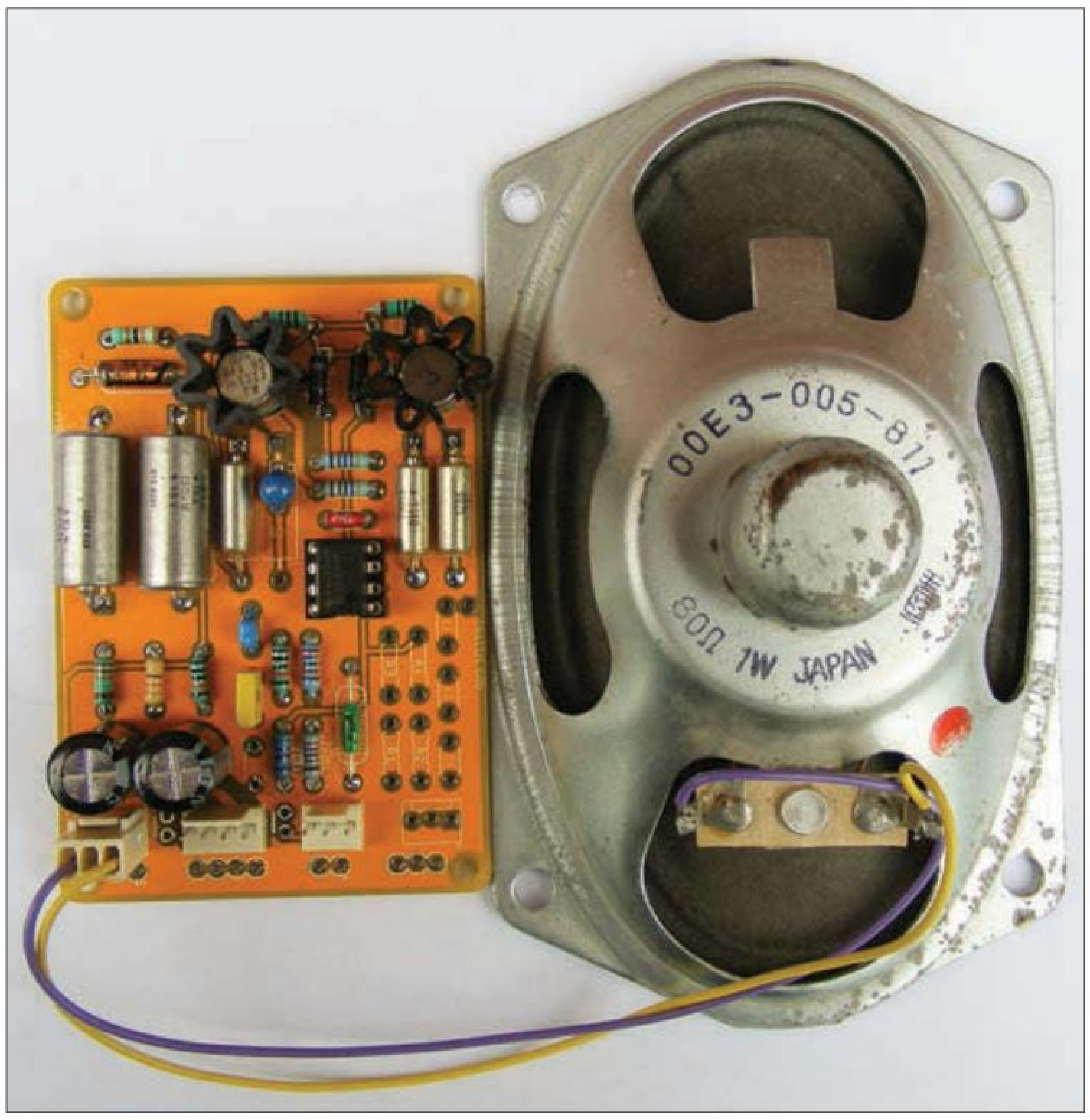


Fig.13. Assembled input *Driver Amplifier* feeding an high-impedance (80Ω) loudspeaker.

a pair of *Driver Amplifiers* to produce a high-voltage bridged power amplifier. (Pads are provided for linking the balanced inputs.) Amplifiers with balanced inputs can be connected in anti-phase to produce a bridged output, as shown in Fig.14. In bridge mode, the real load impedance seen by the whole amplifier is halved because almost double the output voltage swing is achieved. This increases the maximum power by a factor of almost 3.5, so it is essential to have good heatsinks on the output transistors. Since only one bipolar electrolytic output capacitor is required in series with the load, C11 can be linked out of each board.

Note that the phase of one of the boards must be reversed. This can be done by connecting one of the board's balanced inputs in anti-phase (see Fig.15). This gives the benefit of equal input impedance on both input legs, unlike a single differential stage, and this improves interference rejection. Another method is to configure one amplifier as inverting and the other as non-inverting – see Fig.1, last month. Personally, I would always opt for a balanced-input design for lower hum.

Stereo

It is possible to 'piggyback' two *Driver Amplifier* PCBs together for stereo/dual-channel use by linking the power connector, as shown in Fig.16. This amplifier design will also be used in dual-channel configuration for the *Vocoder* input and output amplification, which we will explain later in the series.

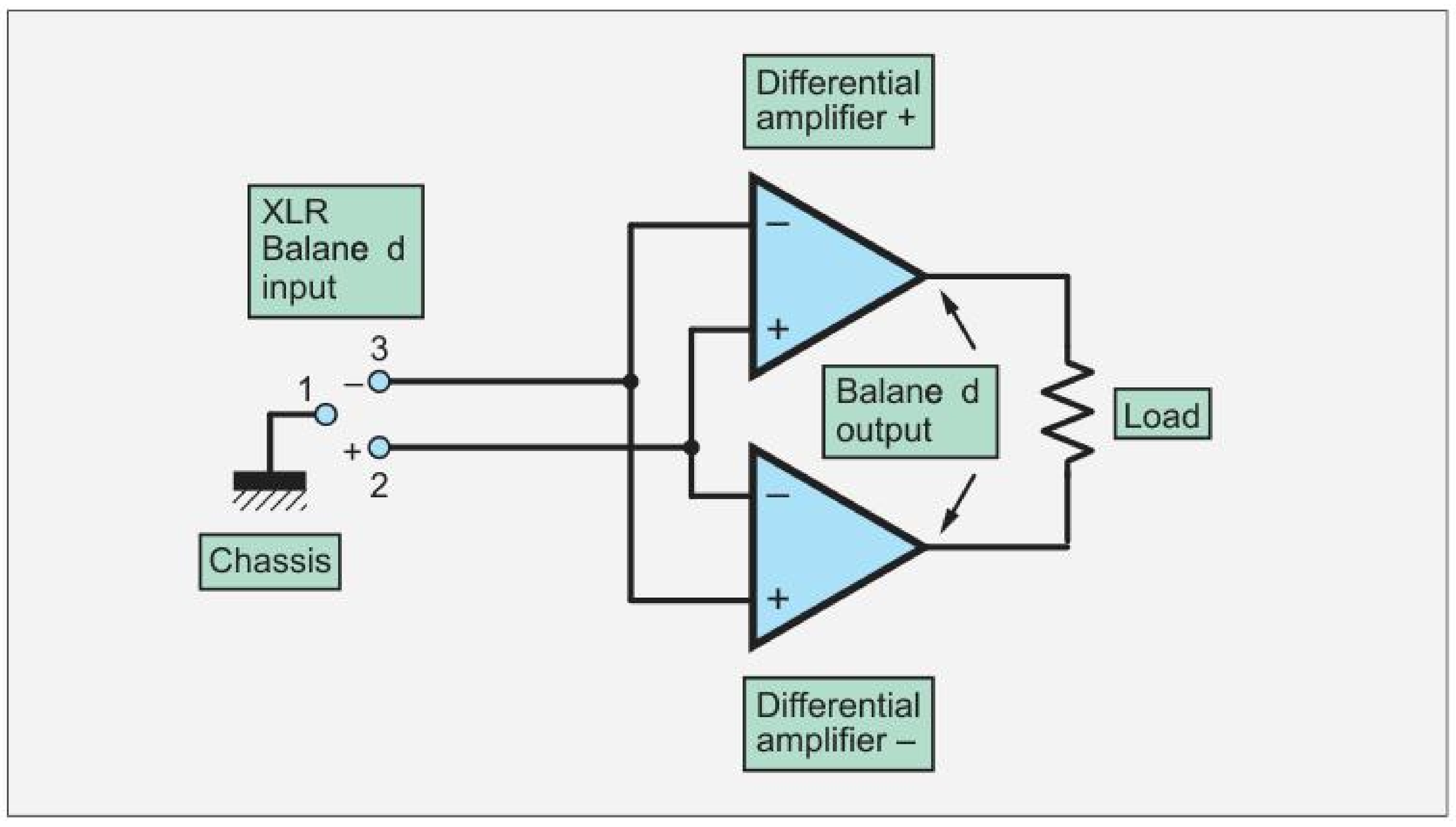


Fig.14. Balanced-bridged driver circuit – useful for balanced-in balanced-out set-ups.

Components list

All parts listed here are for the balanced-input *Vocoder* filter *Driver Amplifier*, the ones marked with a single * are for the *Vocoder* output (summing) amplifier only, and the ones with double ** are for the balanced-input *Vocoder* filter *Driver amplifier* only. (No * means parts are common to both.)

Semiconductors

IC1 NE5534P low-noise audio op amp
TR1 2N2219 NPN medium-power TO5/39 or equivalent, such as BFX85, Rapid order code 81-0116
TR2 BC143 PNP medium-power TO5/39 or equivalent, such as BC161-16 CPC order code SC15338
Any TO5 type with specification: $V_{ce} > 45V$ (60V for OPA604), $H_{fe} > 50$, $I_c > 650mA$, $P_{tot} > 600mW$, $F_t > 50MHz$. (Multicomp Pro make TO5 and TO18 transistors, they are available at a good price from CPC Farnell Preston UK)
D1, D2 Any low voltage small Zener diode – eg, BZY88C2V7

Resistors

All standard audio 0.25W 1% metal-film, such as MSR25. Uncritical positions can be 5% carbon film.

R1, R2	100Ω**
R3, R4, R5*, R6, R7	2.2kΩ
R8	2.2kΩ
R5	100kΩ*
R9, R10, R13	4.7kΩ 5%
R11, R12	10Ω
R14	22kΩ 5%
R15	33Ω 5%
R16	100kΩ 5%

Capacitors

C1**, C2**, C3** C6 C7	470pF 5% 5mm ceramic or film
C12	15pF (typical) NP0 5mm ceramic, see text
C4, C5, C8	22µF 25V
C9	10µF 10V
C10, C11	150µF 16V
C13, C15	470µF 25V
C14	100nF X7R 5mm ceramic

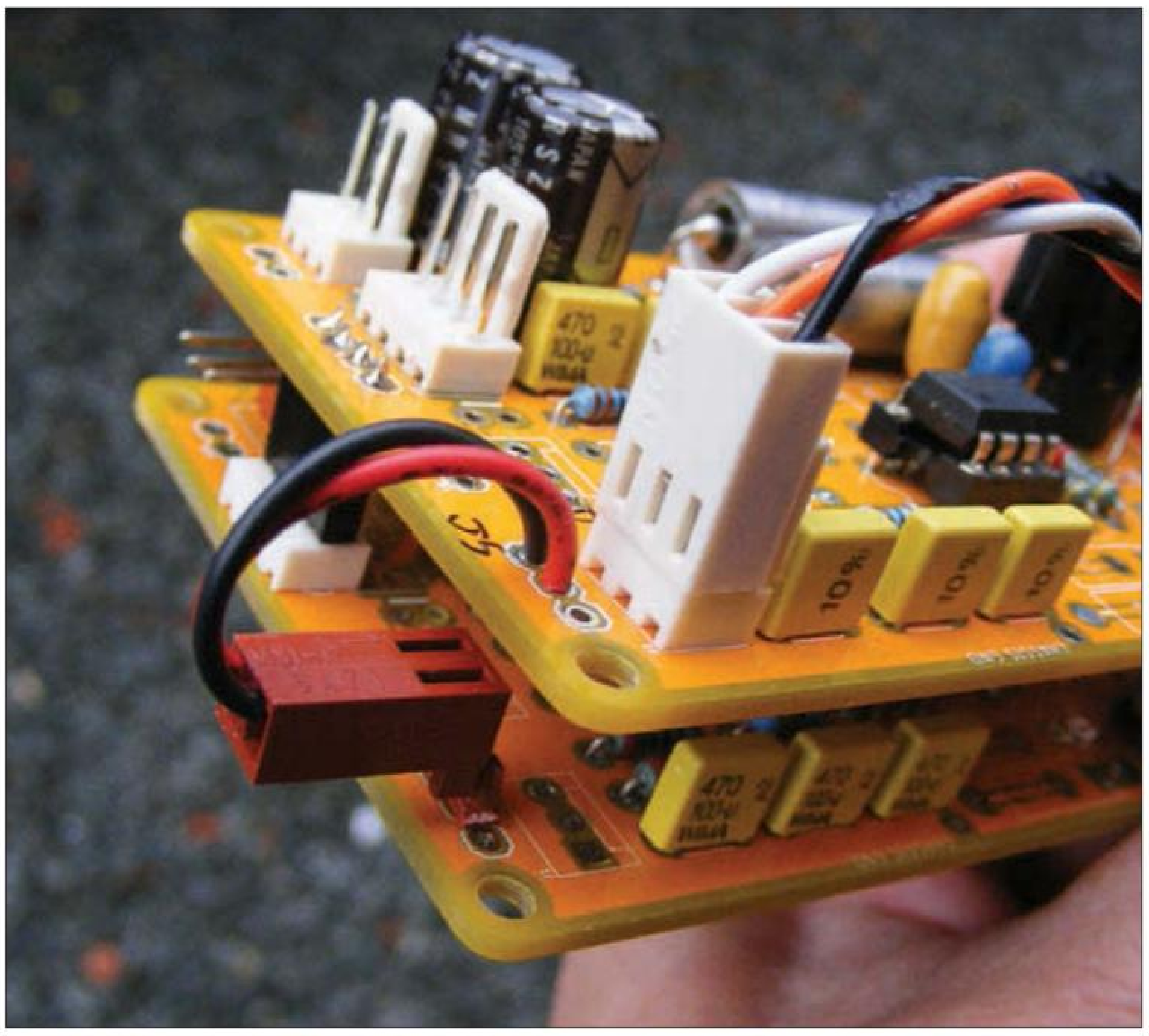


Fig.15. Wiring the inputs in parallel anti-phase on the bridged assembly. Notice the 'flip' on the wires going from J5 on the top board to J5 on the bottom board. This gives an equal input impedance on the 'plus' and 'minus' input pins.

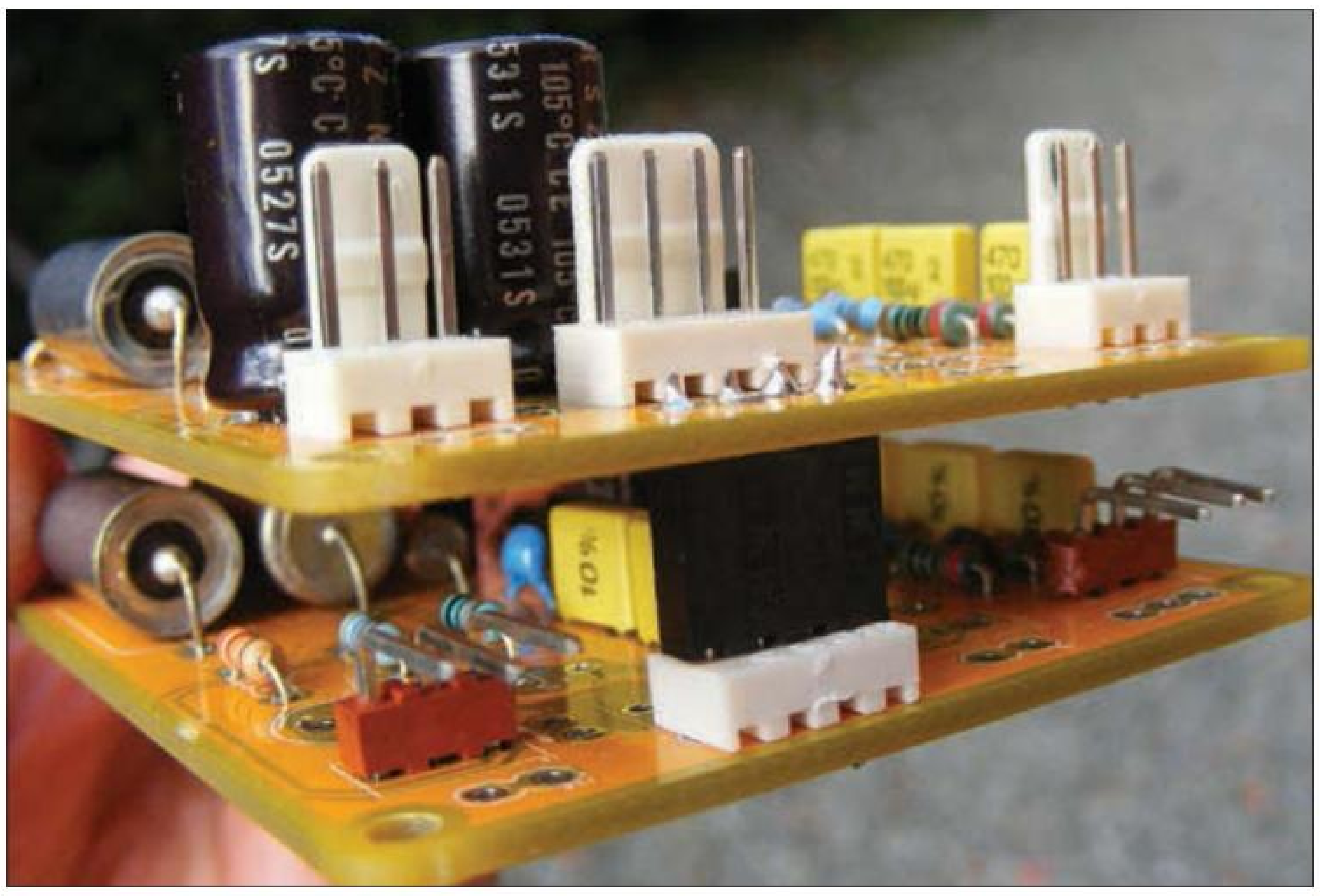


Fig.16. Piggy-backing the power supply connections for stereo.

Note: all electrolytic capacitor positions on the PCB are dual-outline, giving the option of using cheap radial wet aluminium types, or expensive bead/axial hermetic solid-tantalum types for long life. The main decoupling capacitors can also be upgraded to long-life solid aluminium.

Inductors

L1 1mH (only needed for long bus, >150mm) Sigma SC30 or other small axial ferrite type. Epcos B82144A2105J 1000µH ±5% 200mA RF Axial LBC Choke Rapid Order Code: 51-7618

L2 6µH to 15µH low resistance suppressor choke axial. Epcos B82141A1103K 10µH ±10% 410mA RF Axial SBC Choke Rapid Order Code 51-7613

T1 output transformers. Vigortronix VTX-101-3002 or VTX-101-007 Rapid order code 88-2124.

Note these need a few associated passive components, see text.

Miscellaneous

Molex 0.1-inch PCB interconnectors

Transistor mounting pads (2 off) Nylon/PET TO5 RS 402-175, Rapid 38-0278

Heatsinks (2 off) Thermalloy TO5 type 5F 12.5mm 0.5-inch high 45.2 C/W CPC part number SC10803, PR Electronics 300-005.

The transformers and many other hard-to-find audio engineering parts are available from the *AO Shop* – see page 67 in last month's *PE*.

Assembly

When building the circuit, follow the overlay shown in Fig.17. This PCB is a conventional leaded design, so when assembling, just note the points below.

The output transistors are medium-power (metal can) TO5 types. These are more reliable than plastic-cased TO92 parts, such as the BC337/327 normally used for this job. They can also be simply fitted with push-on heat sinks. To ensure a solid mount, a nylon mounting pad should be used, as shown in Fig.18. This needs to be slid over the leads the right way up with the smallest end of the tapered holes towards the bottom. For a solid flush mount, solder one lead, then re-melt it while at the same time pushing the device flat. If it's tilting, then strain to the leads may result in the glass seals cracking. Remember the PNP transistor TR1 goes towards the corner of the board.

To ease heatsink fitting, twisting a flat blade screwdriver in the slot while pushing down helps. This technique requires practice and is a bit risky. The professional technique is to use circlip pliers, as shown in Fig.19. These must be the external type that move *outwards* when squeezed.

The emitter resistors R11 and R12 should be mounted a few millimetres above the board to prevent heat damage to the board in the advent of overcurrent, as shown in Fig.20. This is an important point with all power circuitry where certain resistors can burn under fault conditions.

The Zener diodes D1 and D2 need to be mounted around 1cm high to make contact with the heatsinks. One of the good things about PCBs with plated-trough-holes is that components can be mounted up in the air without the risk of broken solder pads.

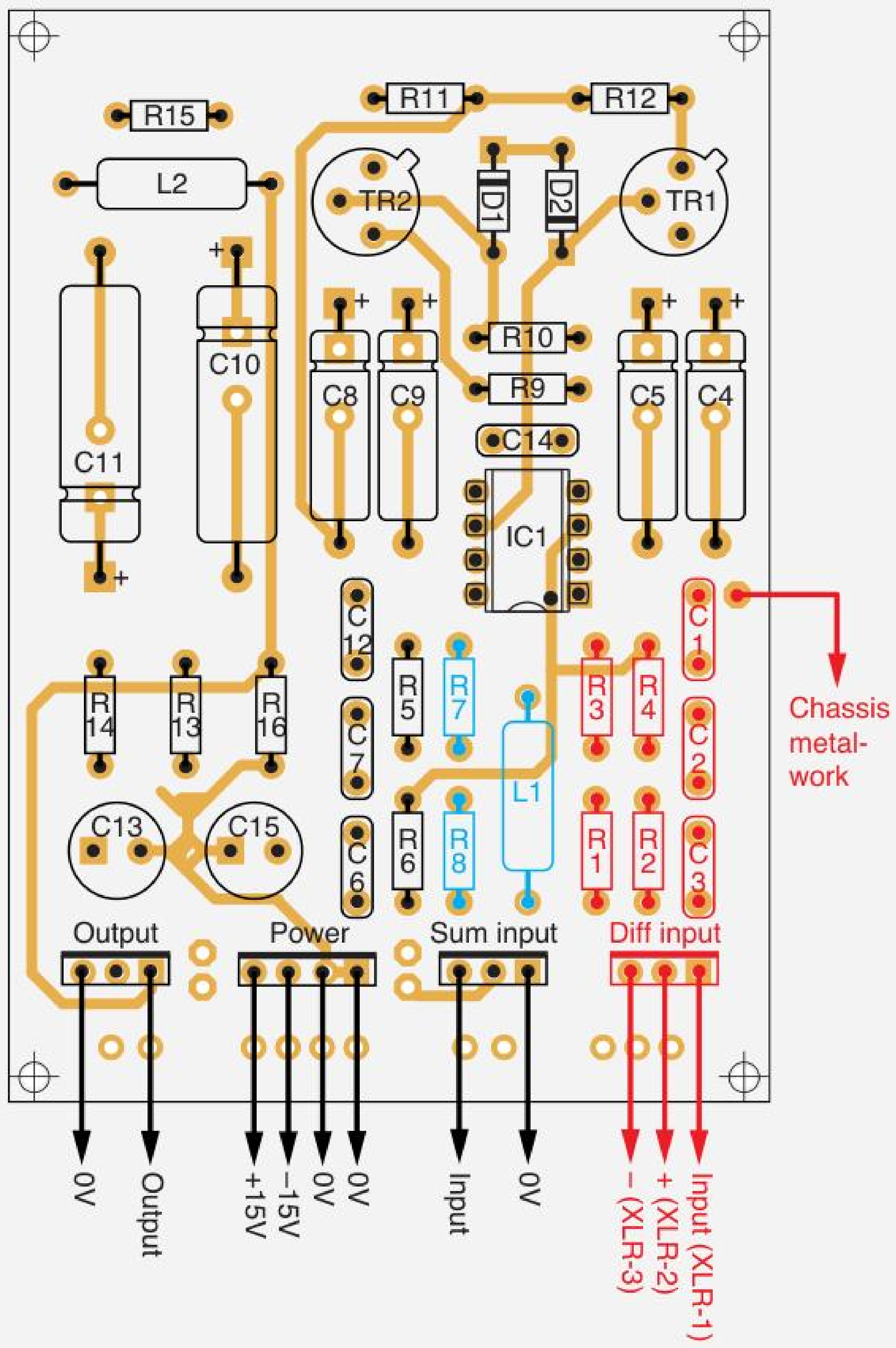

R15
L2
R11
R12
TR2
D1
D2
TR1
C11
C10
C8
C9
R10
R9
C5
C4
C14
IC1
C12
R5
R7
R3
R4
C1
Chassis metalwork
R14
R13
R16
C7
L1
C2
C13
C15
R6
R8
R1
R2
C3
Output
Power
Sum input
Diff input
0V
Output
+15V
–15V
0V
0V
Input
0V
Input (XLR-1)
+ (XLR-2)
– (XLR-3)

Fig.17a. (left) Component overlay for the *Driver Amplifier* module.

Note: the blue components are only for the *Vocoder* output *Driver Amplifier*; the red components are only for the *Vocoder* balanced-input filter *Driver Amplifier*. All other components are common to both.

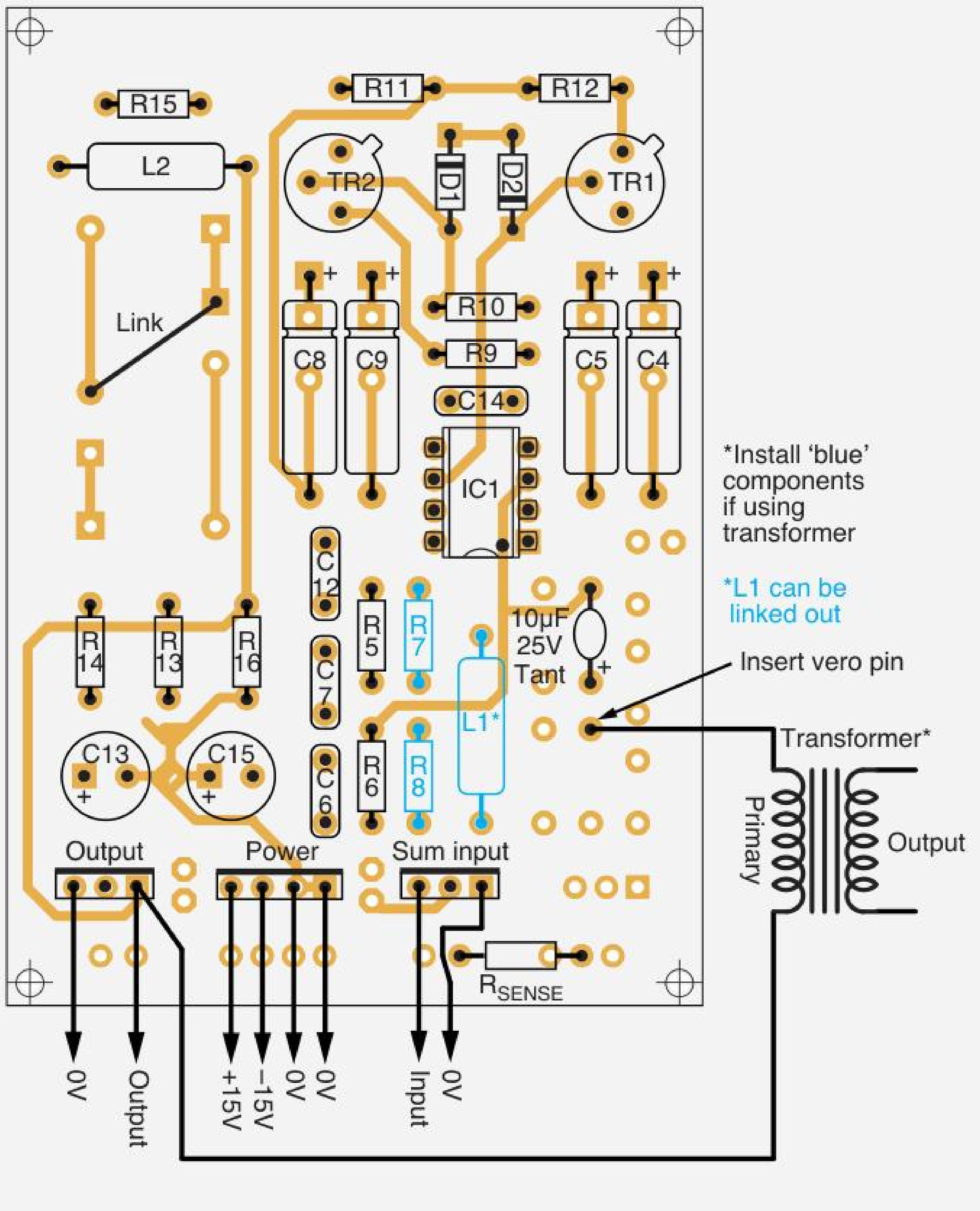

R15
L2
R11
R12
TR2
D1
D2
TR1
Link
C8
C9
R10
R9
C5
C4
C14
IC1
*Install 'blue' components if using transformer
C12
R5
R7
10µF 25V Tant
*L1 can be linked out
Insert vero pin
R14
R13
R16
C7
L1*
Transformer*
C13
C15
R6
R8
Primary
Output
Output
Power
Sum input
R_{SENSE}
0V
Output
+15V
–15V
0V
0V
Input
0V

Fig.17b. (right) Overlay for the low-distortion transformer-output circuit shown in Fig.11.

PCB design: Mike Grindle

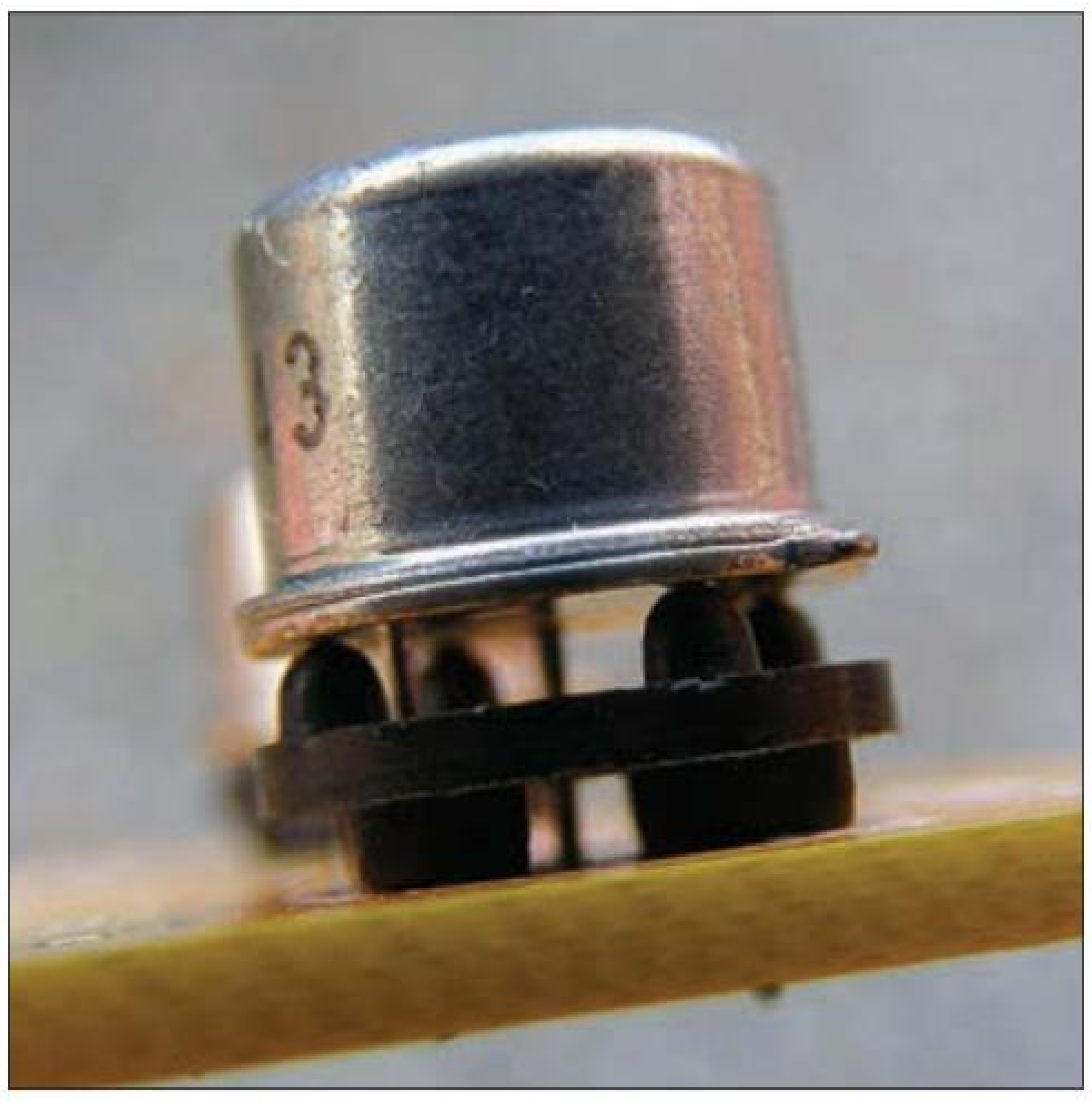


Fig.18. Plastic mounting pad for supporting a TO5 transistor.

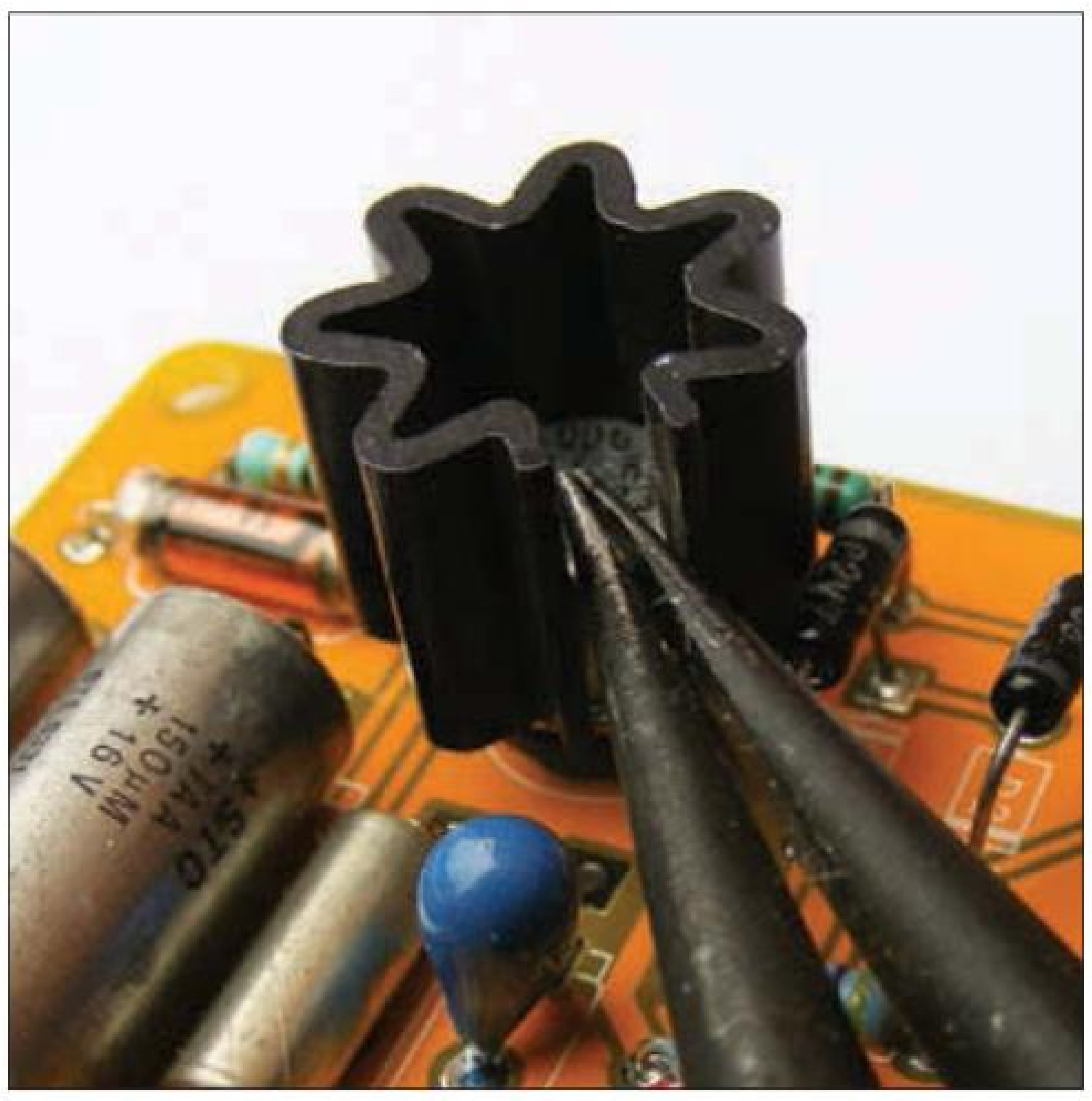


Fig.19. Fixing the heatsink of a TO5 with circlip pliers.

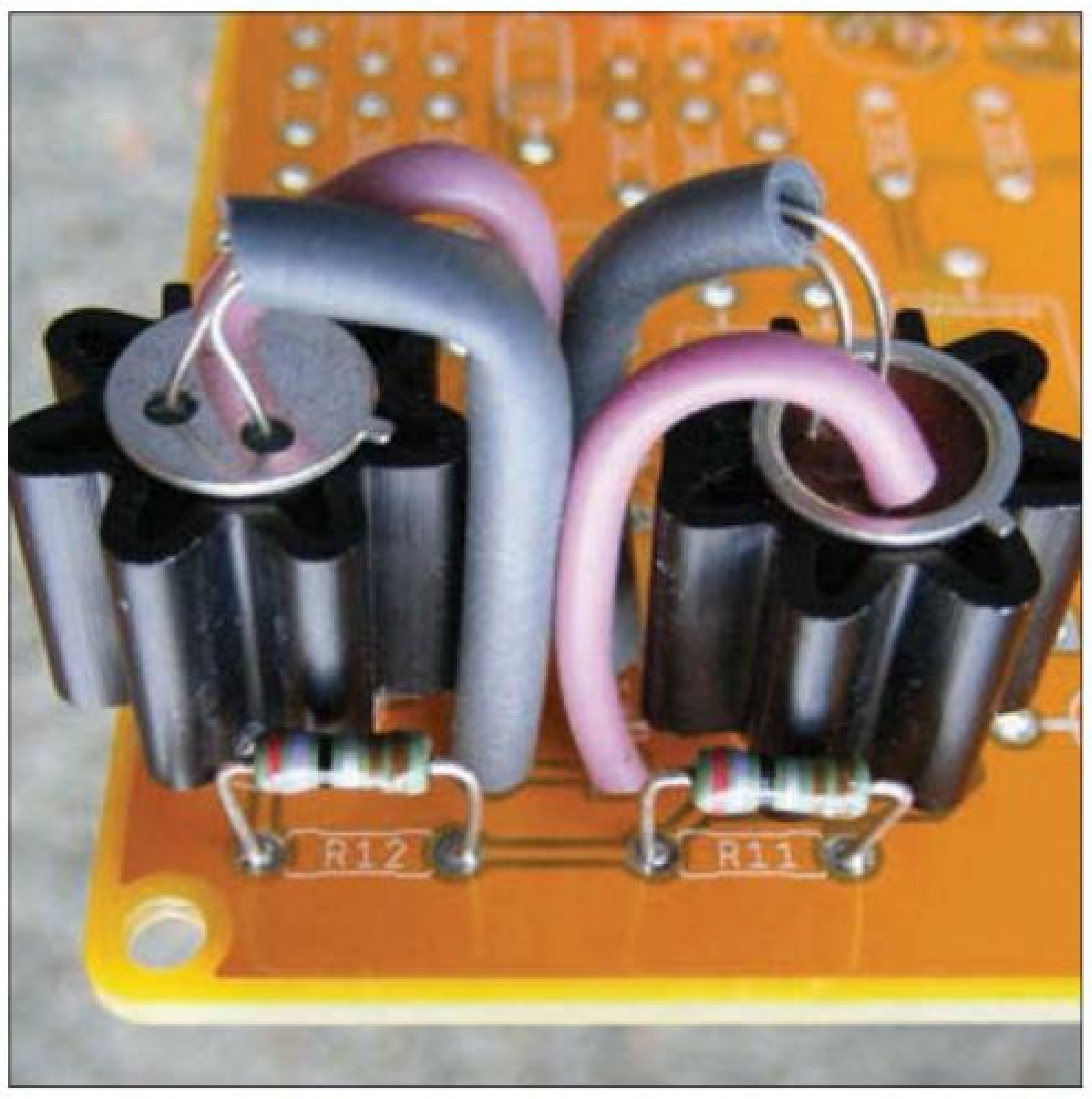


Fig.20. Emitter resistors R11/R12 mounted off the board to avoid heat damage to PCB laminate. Note the transistors used experimentally as bias diodes. These must be insulated since TO5 transistor cases are connected to their collectors.

Testing

Once you are happy with assembly and have double checked all parts for value, location, orientation and solder bridges, it's time for testing. Turn on for a short period, make sure the current on each rail is below 30mA and nothing gets hot. The emitter resistors R11 and R12 will smoke if something is amiss, such as a polarised device inserted incorrectly, causing excess current. Left for more than a few seconds in this state and the transistors will join the resistors in the component graveyard. Since the op amp can withstand shorts to either power rail for a short period, it usually survives. Check the output offset voltage at the junction of the emitter resistors. Ideally, it should be less than 0.1V.

Check the quiescent current by measuring the voltage across R11 and R12. It should be 0.1V. If it is much higher than this, then the quiescent current is too high. Occasionally, with transistors having an exceptionally low turn-on voltage or V_{ce}, the emitter resistors will have to be increased. TO5 silicon planar devices typically range from 0.65 to 0.72V, so there should be no problems.

Another useful check is to look for high-frequency (hundreds of kilohertz upwards to megahertz) output instability with an oscilloscope. This is an essential check if unusual reactive loads, different op amps or slow (F_t < 20MHz) output transistors are used.

Distortion

Distortion tests require expensive test gear, but it is not a worry for *Vocoder* use since the distortion from the VCAs is in the order of a few percent and swamps any amplifier-related distortion. However, if you want to use this amplifier with Hi-Fi then it's vital to get the distortion as low as possible because op amp high-order, odd-harmonic distortion is quite unpleasant. All of the following commentary is related to Hi-Fi, not *Vocoder* applications.

If you are lucky enough to have a Marconi, Wayne Kerr AMS1, Radford or Audio Precision analyser you should be able to get some pretty graphs.

This is a simple clean amplifier, so the distortion and frequency response curves can be rather boring. I'm not going to bother printing the frequency response because it's just a flat line from 20Hz to 20kHz!

Op amp selection

Let's first look at op amp choice – I tried several audio-quality devices. The ever-popular NE5534 gave the lowest distortion with the input balanced filter *Driver Amplifier*, at 0.001% – despite being a 1970s design. The OPA604 and the venerable LM318 gave slightly more distortion at 0.0015%. The LM318 was *the* op amp for low-distortion audio before the revolutionary TDA1034/5534 came out. However, it was noisy, at 50nV/√Hz. (True to form, the distortion was lower in the inverting amplifier configuration, by about half.) Here, the OPA604 was the winner with 0.0004% distortion and noise (THD+N).

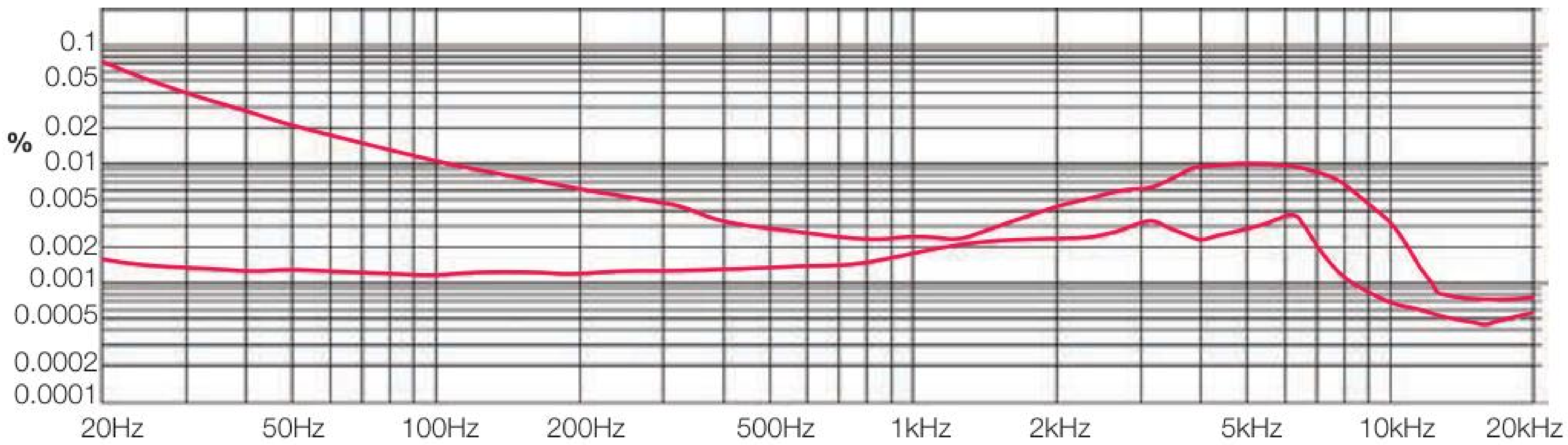


Fig.21. (above) Distortion with the VTX-101-3002 transformer and (below) the result of negative output resistance using circuit in Fig.11. Here R_{SENSE} = 18Ω, gain = 4×.

Ouput capacitor

If you're using our *Driver Amplifier* to supply an input to a Hi-Fi amplifier, then it must deliver zero DC offset. A good way to do this is by adding biased capacitors to the *Driver Amplifier's* output (C10/C11). In this circuit, tests showed that the addition of this tantalum output capacitor network only doubled the very low distortion over 20-30Hz.

The effect of load

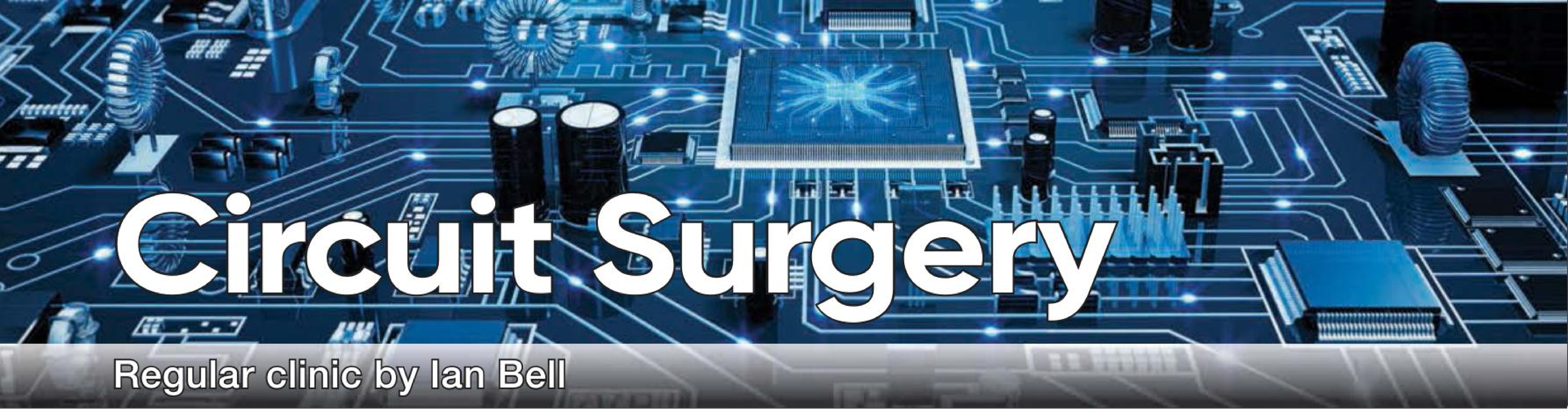
Next, Let's look at loading effects. Reducing the *Driver Amplifier's* load from 600Ω to 50Ω to simulate headphones or a speaker increases distortion a small amount, but it was still below the audibility threshold.

In the same 50Ω loading situation, the TL071 op amp had insufficient drive capability and was therefore rejected.

When adding the output balancing transformers, the distortion increases, especially at low frequencies. Fig.21 compares distortion levels with and without the negative-output-resistance, distortion-reduction technique discussed above. This graph used the Vigortronix VTX-101-3002 transformer. I found that a cheaper transformer, the VTX-101-007, can give very good distortion, but only at low levels, around 0dBm. This cheaper transformer is perfect for the *Vocoder*.

Next month

We are roughly halfway through our all-analogue *Vocoder* project. Up next, we will discuss and build the two types of the all-important *Vocoder* filter boards – high/low pass and bandpass.



Circuit Surgery

Regular clinic by Ian Bell

Op amp offsets – Part 1

On the EEWeb Forum, user Deion posted about precision op amps, first quoting the OP97 datasheet from Analog Devices: 'The OP97 is a low-power alternative to the industry-standard precision op amp, the OP07. The OP97 can be substituted directly into OP07, OP77, AD725, and PM1012 sockets with improved performance and/or less power dissipation and can be inserted into sockets conforming to the 741 pinout if nulling circuitry is not used. Generally, nulling circuitry used with earlier generation amplifiers is rendered superfluous by the extremely low offset voltage of the OP97 and can be removed without compromising circuit performance'. Then Deion asks 'What confused me is the range of temperature? Should the resistor be balanced?'.

Questions

We are not sure exactly what Deion means in asking about the temperature range, as with the reply on the forum, the obvious answer is from the first page of the datasheet (and repeated in the 'absolute maximum ratings' table) – the device has an 'extended industrial temperature range' of –40°C to +85°C. However, specifications for bias current and various graphs of device characteristics on the datasheet cover a larger range of –55°C to +125°C. The –40°C to +85°C specification is an 'absolute maximum rating' which means that the device may be damaged outside this range or become unreliable if used close to the specified limits for extended periods. However, the op amp may work (at least for a while) beyond these stress ratings.

The second question concerns the balancing of resistors in precision op amp circuits – the purpose of this is to reduce offsets due to bias currents flowing in the external circuit (although it is not always appropriate). In this month's article, after discussing the basic concept of offsets, we will focus on op amp internal offsets, in particular looking at how input offset voltage is defined and modelled. Next month, we will look at input bias currents, and address the issues related to resistor balancing.

Offsets

In simple terms, offsets are DC errors in a circuit's output due to imperfections in the circuit or components. This issue is important in DC and very-low-frequency circuits, a key example being amplifiers used for sensor signals (such as temperature) where the quantity being measured changes slowly – here offsets cause inaccuracies in the measurement.

In other cases, unwanted DC output may be damaging to a load intended to only be driven by an AC signal with zero offset (eg, a loudspeaker). Unwanted DC can be blocked using coupling capacitors, but this may not be practical if low wanted-signal frequency leads to a requirement for very large capacitance values. Even if available, capacitors with suitable values may be physically too large, or too expensive or have non-ideal characteristics of their own which affect the signal in an unacceptable way.

If offsets were pure DC they would be fixed for all time and could be removed by a one-off calibration/cancellation procedure, but real offsets drift due to changes in temperature, aging and other factors that influence the circuit. These changing offsets are just like low-frequency signals and so are amplified along with the wanted signal – they act as low-frequency noise.

Precision

If offsets are changing in the same frequency range as the wanted signal, then it is not possible to use coupling capacitors to block the offsets, irrespective of the capacitors' characteristics or cost (they would block the signal too). The only solution is to design circuits with inherently low offsets. As with any engineering design, making one aspect better tends to be at the expense of something else, for example lower offsets may mean poorer high-frequency response or higher power consumption.

Op amps with low offsets as a key characteristic are often referred to as 'precision' op amps, like the OP97 mentioned in Deion's post. The best offset performance is obtained by auto-zeroing amplifiers or chopper-stabilised amplifiers, but standard precision op amps can also provide good performance and may be more suitable in some applications, or where the precision requirements are not so stringent. We are focusing on standard precision op amps in this article.

In an op amp circuit, offsets at the outputs can occur due to offsets in the op amp's internal circuitry, or due to the external circuitry and its interaction with the op amp. The resistor balancing mentioned by Deion is an example of the latter situation, which we will discuss next month.

Input offset voltage

For an op amp, ideally, with a differential input of zero, the output should also be zero, but with real op amps there will typically be a non-zero output. The input offset voltage (V_{IO}) is defined as the DC voltage which must be supplied between the inputs to force the quiescent (zero input signal) open-loop (no feedback applied) output voltage to zero. The definition of input offset voltage is illustrated in Fig.1. Typical values for standard (not autozeroing/chopping) precision op amps range from about 10μV to 500μV. Input offset voltage for an individual device can be of either polarity up to the specified value.

Temperature often has a significant effect on offsets – the temperature coefficient of input offset voltage specifies how V_{IO} changes with temperature, typical values for standard precision op amps are around 0.1 to 10μV/°C. The datasheet for an op amp may also have a graph showing offset variation with temperature. Aging of devices also causes offsets to vary. Again, this will often be specified

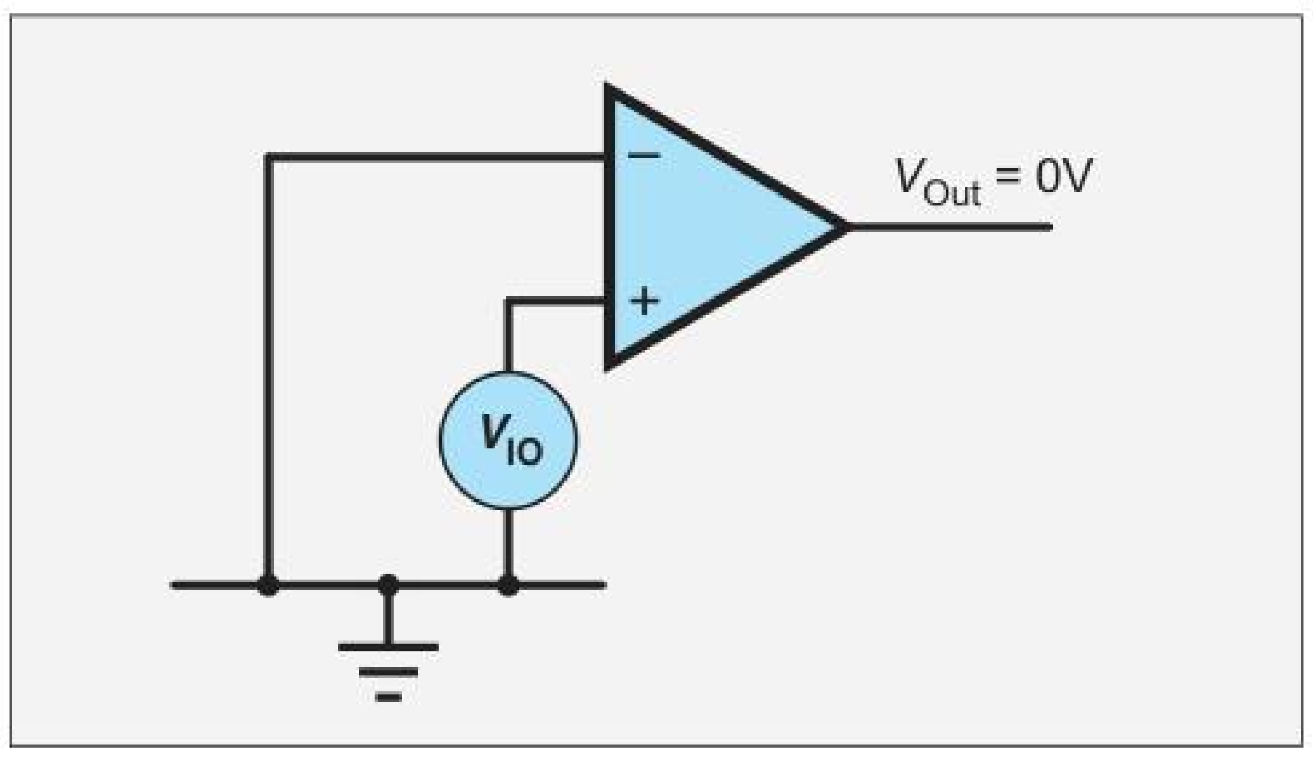


Fig.1. Offset voltage defined.

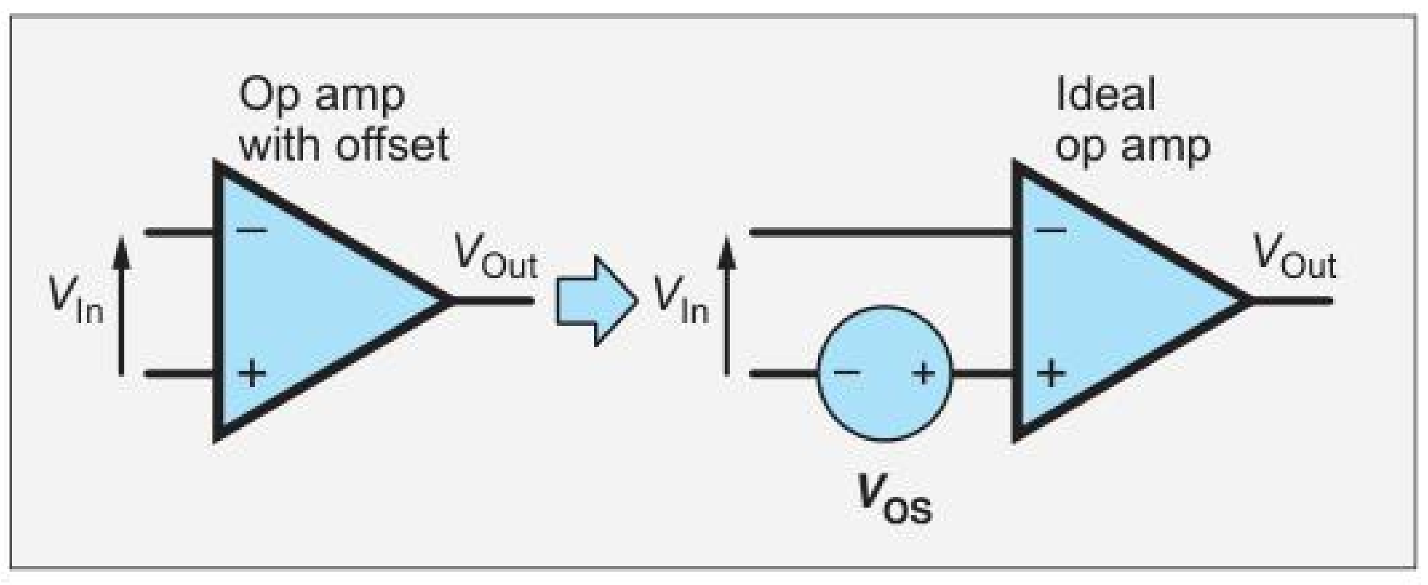


Fig.2. An op amp with offset can be modelled as an ideal op amp with an offset voltage source at its non-inverting input.

on the datasheet, for example, as a long-term stability value in µV/month.

The circuit in Fig.1, used to define input offset voltage, can also be the basis for analysing the offset in a circuit. For analysis purposes we can replace an op amp with offset with an ideal op amp plus an offset voltage source, as shown in Fig.2. The offset represented in this way is called 'input referred'. Although we are discussing input offset voltage, other sources of unwanted output from the op amp, such as random noise and the op amp's response to power supply voltage variation can also be represented in this way.

Modelling

If an input, V_{In} is applied to an op amp its output will depend on both the input signal and the input offset voltage. For the open-loop amplifier circuit in Fig.2, the output will be

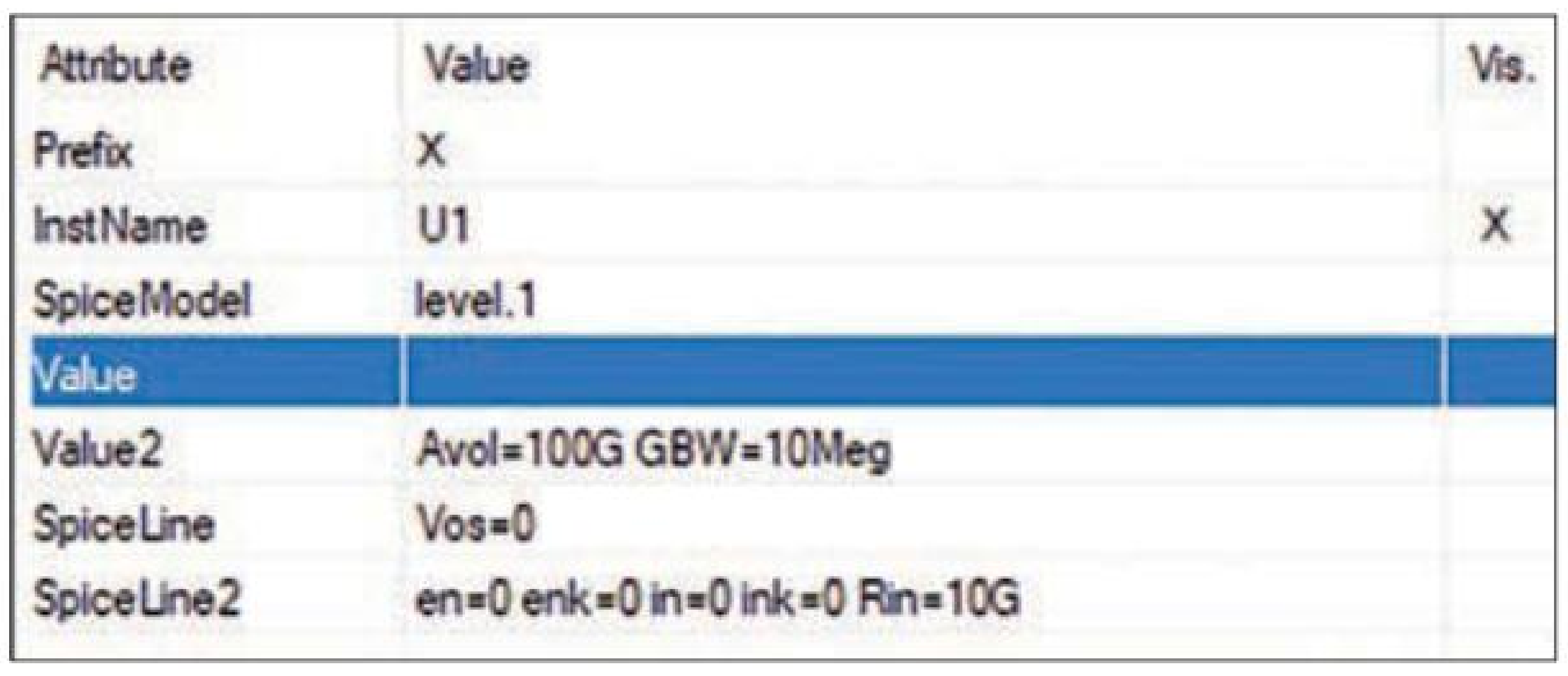
$$V_{Out} = AV_{In} + AV_{OS}$$

Where A is the open-loop gain of the op amp. The value AV_{OS} is the 'output referred' offset. Normally, we do not use an op amp open loop, in which case the effect of the offset voltage can be analysed in the context of the specific circuit using the approach in Fig.2. The input-referred offset does not change, but the output referred offset will depend on the circuit in which the op amp is used. In a circuit with feedback the output offset depends on the circuit, not on the op amp's open-loop gain.

Use of a voltage source connected to the non-inverting input to model offset can be investigated in LTspice, but we need to use an idealised op amp with zero internal offset to do this. If we select a real op amp (we often do this for *Circuit Surgery* simulation examples), it will model that device's offset, which we will not be able to control – which is not what we want here. Instead, we can use the `UniversalOpamp2` component from LTspice's library – it is right at the end of the list in the Op amps section of the Component Selection menu. An LTspice schematic with an inverting and a non-inverting standard op amp amplifier – both using `UniversalOpamp2` – is shown in Fig.3. This circuit includes voltage sources, as discussed above, to model the offset voltage.

UniversalOpamp2

You may notice there are two other components that are not real op amps on the component menu: `opamp` (without supply connections) and `opamp2` (with supplies). These can also be used to include idealised op amps (or real op amps) on a schematic but require an LTspice subcircuit to define their internal circuit or behaviour. `UniversalOpamp2` is more convenient for working with generic op amp behaviour because it can be configured by right-clicking the op amp symbol and setting the various parameters (if values other than the defaults are needed) via the Component Attribute Editor – see Fig.4. Note the offset parameter (`Vos`) for the `UniversalOpamp2` is set to zero in our simulation because we are using the external sources `VOS1` and `VOS2` to model the offset.



Attribute	Value	Vis.
Prefix	X	
InstName	U1	X
SpiceModel	level.1	
Value		
Value2	Avol=100G GBW=10Meg	
SpiceLine	Vos=0	
SpiceLine2	en=0 enk=0 in=0 ink=0 Rin=10G	

Fig.4. Using the LTspice Component Attribute Editor to configure the UniversalOpamp2 component.

`UniversalOpamp2` can model op amps with different amounts of detail. Setting the `level` value for the component determines which model is used. There are four levels (1, 2, 3 and 3a in increasing levels of detail). For information on all the parameters refer to the example schematic installed by LTspice (typically at location **...\Documents\LTspiceXVII\examples\Educational\UniversalOpamp2.asc** on Windows). Use of idealised models with certain specific non-ideal characteristics allows us to explore the effect of those characteristics in isolation from other effects to gain insights about circuit behaviour.

For this example, we will use the simplest (level 1) `UniversalOpamp2` model. This does not use the power supplies (eg, to limit the output voltage) but power supplies have been included on the schematic to allow the model level to be switched easily if required. We need near-ideal behaviour, so we have set the open-loop voltage gain (`Avol` model parameter) very high (100G = 1×10^{11}) so any errors in output voltage due to non-ideal (non-infinite) open-loop gain are not significant. Similarly, the input resistance (`Rin` model parameter) is set very high (10GΩ) so there is little impact on the currents in the external resistors.

Simulation

The op amp amplifiers in Fig.3 have the same resistor values. Using the well-known formulae for the gain of these circuits we get the following. For the inverting circuit, the gain is: $-R_{F1}/R_{I1} = -50k\Omega/1k\Omega = -50$. For the non-inverting circuit, the gain is $1 + R_{F2}/R_{I2} = 1 + 50k\Omega/1k\Omega = 51$.

With 10mV input provided by source `V1` we would expect outputs (with no offset) of $-50 \times 0.01 = -0.5$V for the inverting circuit and $51 \times 0.01 = 0.51$V for the non-inverting circuit. However, the circuit includes offsets, introduced by `VOS1` and `VOS2`.

We just need to find the DC output, so we can use an operating point simulation (.op SPICE directive). The results are:

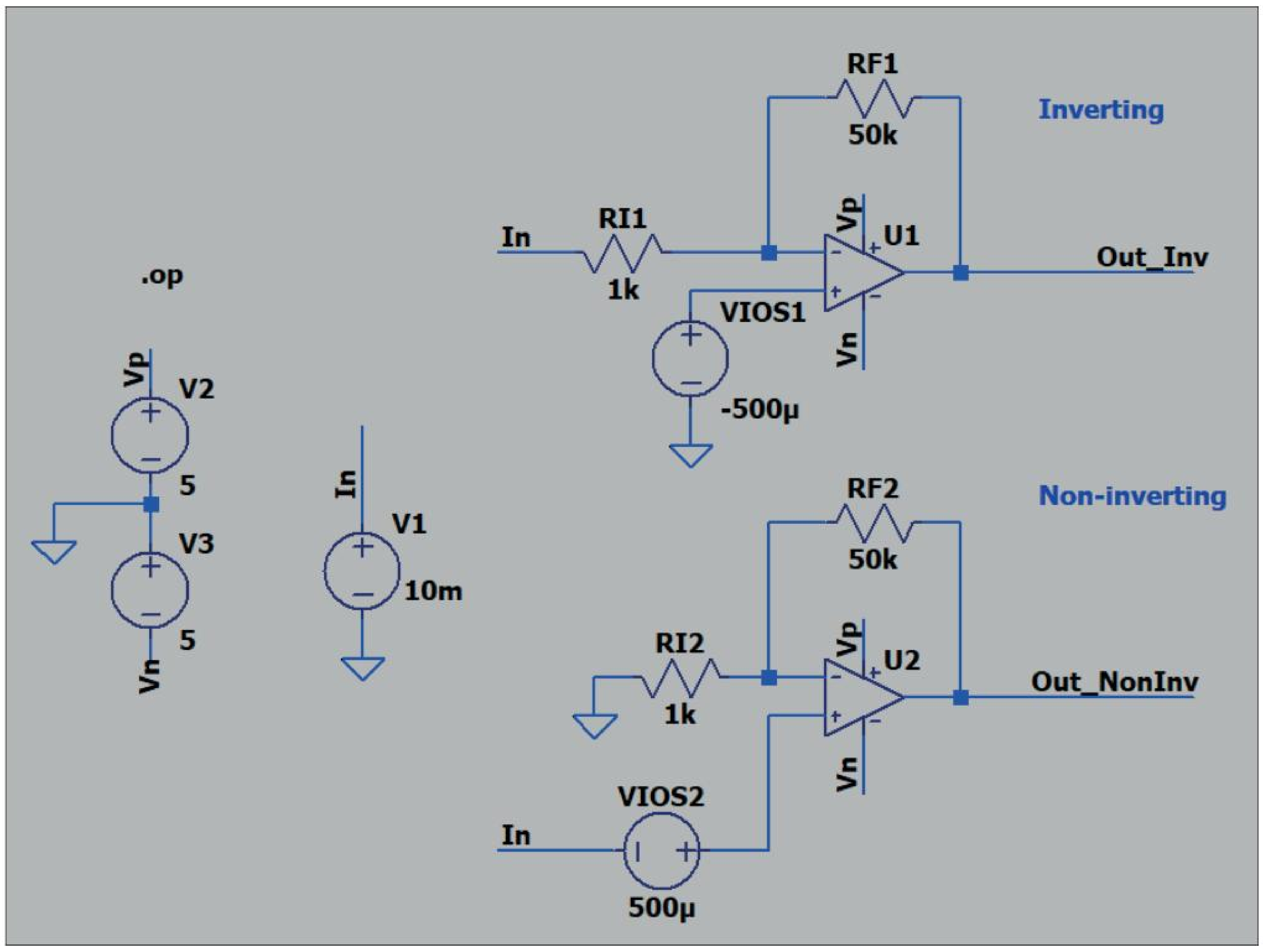


Fig.3. LTspice schematic for simulation to investigate modelling of offset voltages.

```
--- Operating Point ---

V(in):          0.01      voltage
V(n003):        0.0105    voltage
V(out_noninv):  0.5355    voltage
V(vp):          5         voltage
V(vn):          -5        voltage
V(n001):        -0.0005   voltage
V(out_inv):     -0.5255   voltage
V(n002):        -0.0005   voltage
V(n004):        0.0105    voltage
```

In both cases the output voltages `V(out_inv)` and `V(out_noninv)` are shifted by 22.5mV from the values obtained in the calculation above. For the inverting circuit we have –0.5 – 0.0255 = –0.5225 and for the non-inverting 0.51 + 0.0225 = 0.5355. The magnitude of the offset at the output (with the same input offset) is the same for both circuits despite the different circuit gains. This is because the offset model is applied directly to the non-inverting input in both cases. As far as the offset voltage sources are concerned, both circuits behave as non-inverting amplifiers. We can see this using some circuit theory called the 'superposition theorem' – if we have a linear circuit with multiple voltage and current sources the output is equal to the sum of effects of the individual sources, with the others set to zero.

For the circuit in Fig.3, if we set the offset sources to zero (as for an ideal op amp) we get the standard amplifier circuits for which we have just calculated the output. We can also consider just the effect of the offset sources by setting the input voltage to zero (equivalent to shorting the input to ground). Doing this results in the circuits shown in Fig.5. We have the same circuit in each case – a non-inverting amplifier with the offset source as the input. Thus, in both cases the output due to the offset will be the non-inverting gain times the input offset voltage, in this case: 51 × ±500µV = ±22.5mV.

The sign of the offset was set in the circuit of Fig.3 to increase the magnitude of the output, to make the result more obvious. Changing the polarity of the offset sources would reduce the magnitude of the total output in both cases (offset opposite in polarity to wanted output), but the offset magnitude would be the same.

Noise gain

The fact that offsets, and random voltage noise (which is modelled as input referred to the op amp's input in the same way), are amplified by the circuit's non-inverting gain, irrespective of the amplifier circuit configuration, leads to the idea of 'noise gain'. The noise gain of the circuit is the gain which applies to a voltage applied directly to the op amp's inputs. It is equal to the non-inverting gain. The noise gain, rather than signal gain, is used when calculating output offset (or noise) for an op amp circuit. The term is commonly used in technical documents by op amp manufacturers discussing offsets and noise.

Although the offset (or noise) voltage source is commonly depicted connected to the non-inverting input, moving it to the inverting input does not change the fact that offset (or noise) is amplified by the noise gain (non-inverting gain). This can be verified by the LTspice circuit in Fig.6, which produces the same output voltages as the circuit in Fig.3. This may not be obvious at first sight, but consider the fact that in a feedback circuit the op amp is maintaining (in the ideal case) zero volts between its inputs, so moving the source past the inputs does not change the effect it has on the output.

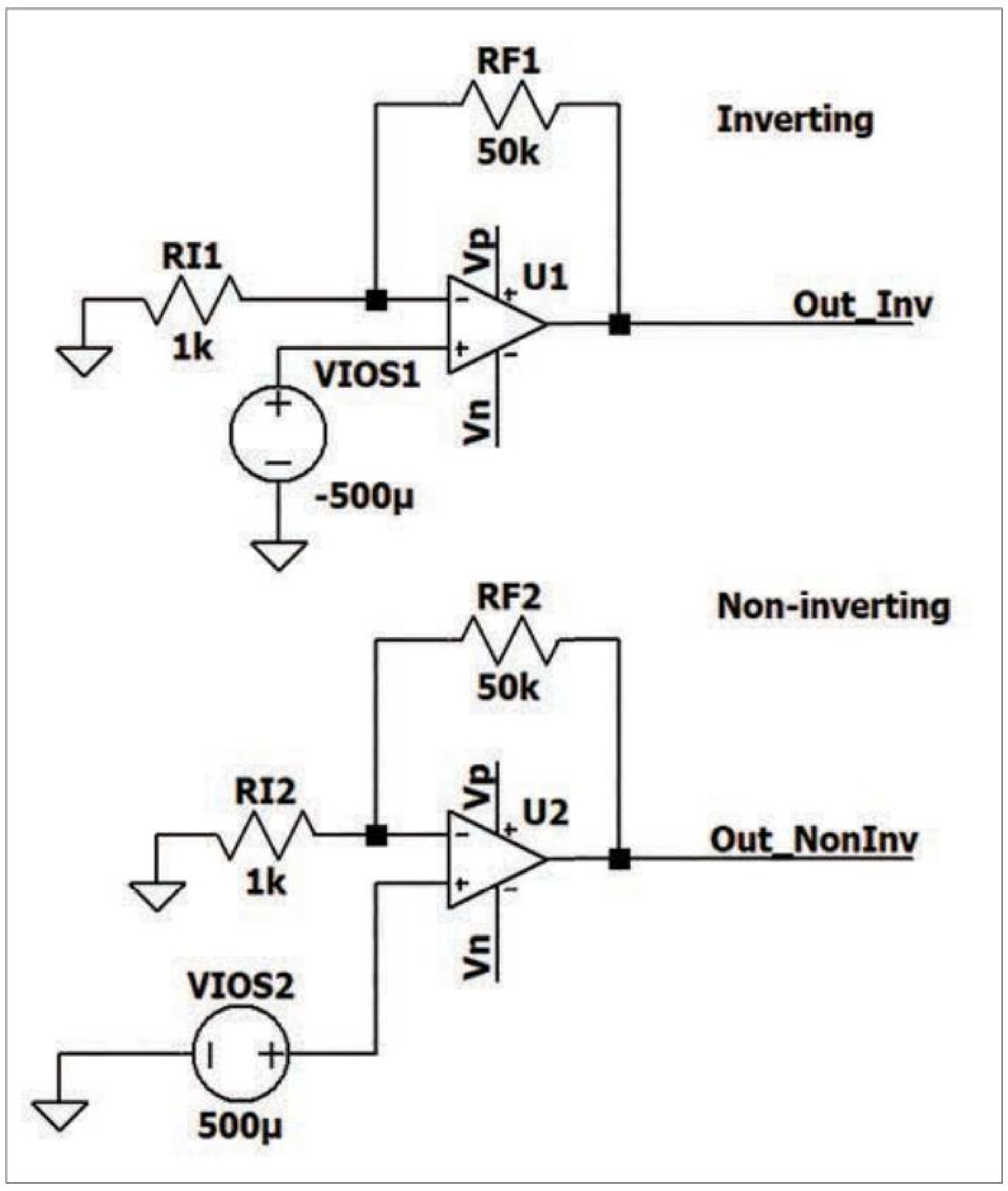


Fig.5. If we consider just the contribution of the offset for two amplifiers in Fig.3, then we find that in both cases offset is amplified by the non-inverting gain.

Another way to understand that the offset is not amplified by the circuit gain in the inverting amplifier is to consider the fact that in the inverting circuit (in Fig.6) the offset source is connected directly to the op amp input, so it is not going to be affected by the resistors in the same way as the input signal. The input signal is attenuated by the potential divider formed by R_{I1} and R_{F1} before arriving at the op amp (by a factor of 50kΩ/(50kΩ+1kΩ) = 50/51 = 0.9804 in our example). The magnitude of the gain of the inverting circuit is smaller than the non-inverting gain by this factor (50 = 51×(50/51)).

Offset nulling

Even with a low offset, op amp users may want a circuit to have the facility to manually adjust the offset to minimise (or null) it. This feature is provided by some op amps, including the OP97, despite as quoted above, the datasheet indicating that it is 'rendered superfluous'. Typically, if offset adjustment is available, the op amp will have two pins labelled 'Null' (see Fig.7 for the OP97 pinout).

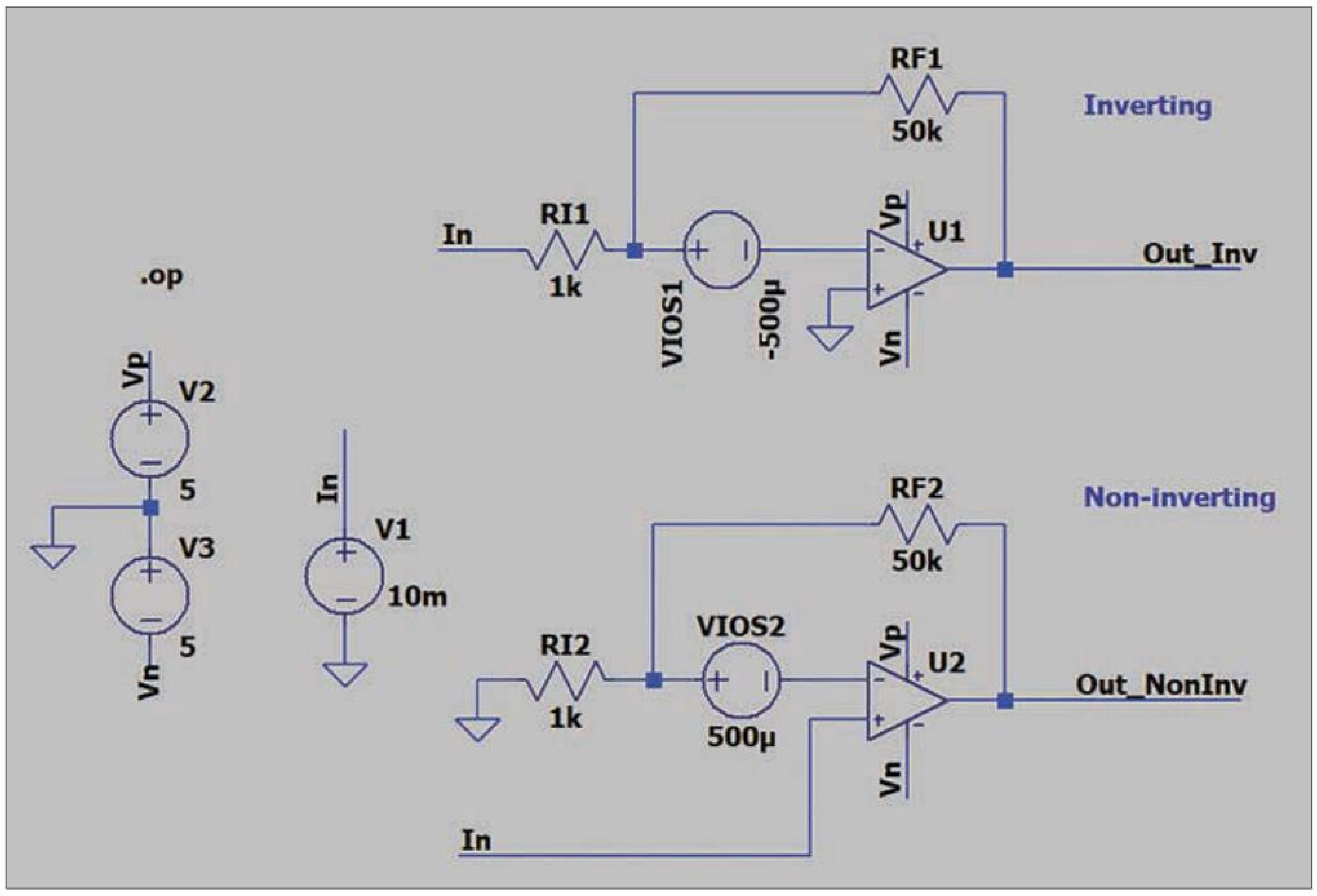


Fig.6. Moving the offset voltage source to the other input produces the same results.

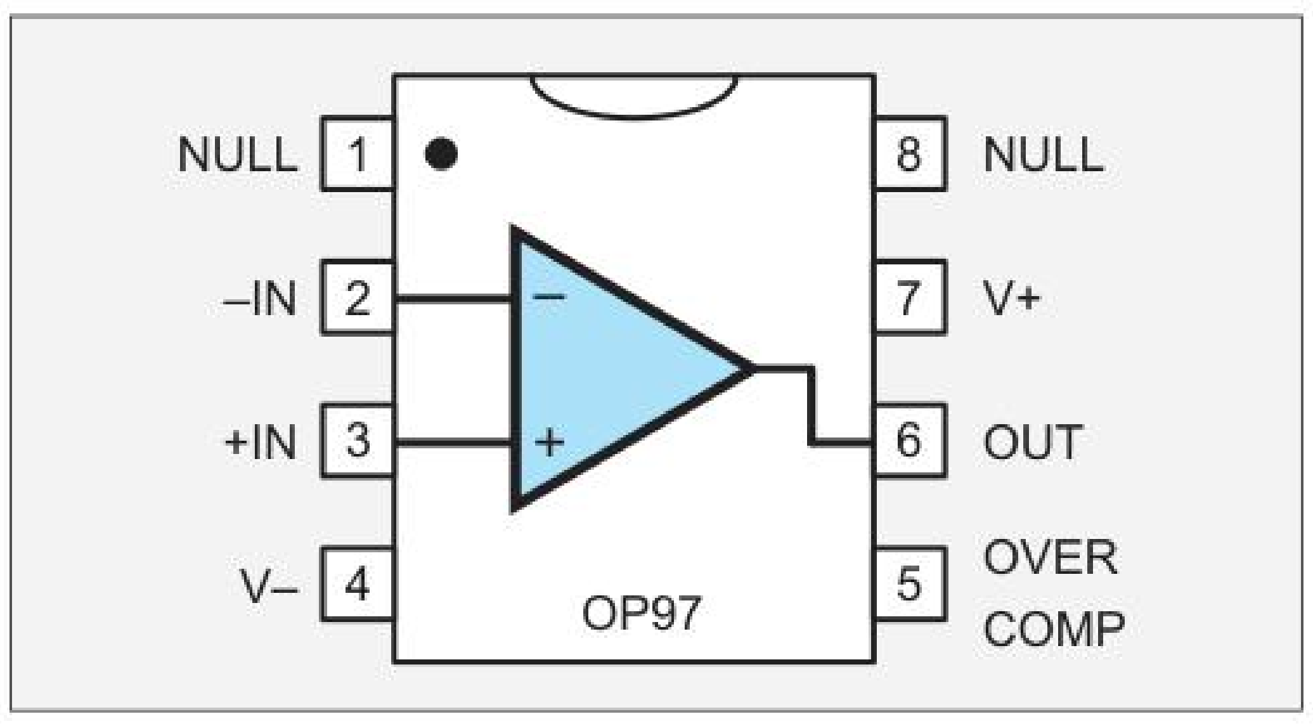


Fig.7. OP97 op amp pinout.

Typically, to use the null pins they are connected to a trimmer potentiometer, with the wiper going to one of the supplies, as shown in Fig.8. For the OP97 it is the positive supply. For the 741 it is the negative supply – hence the comment in the quote above that the OP97 'can be inserted into sockets conforming to the 741 pinout if nulling circuitry is not used'. If an op amp is swapped with one using a nulling circuit connected to the opposite supply, it is likely to be destroyed. The device datasheet must always be checked for the exact details of using nulling pins where they are available.

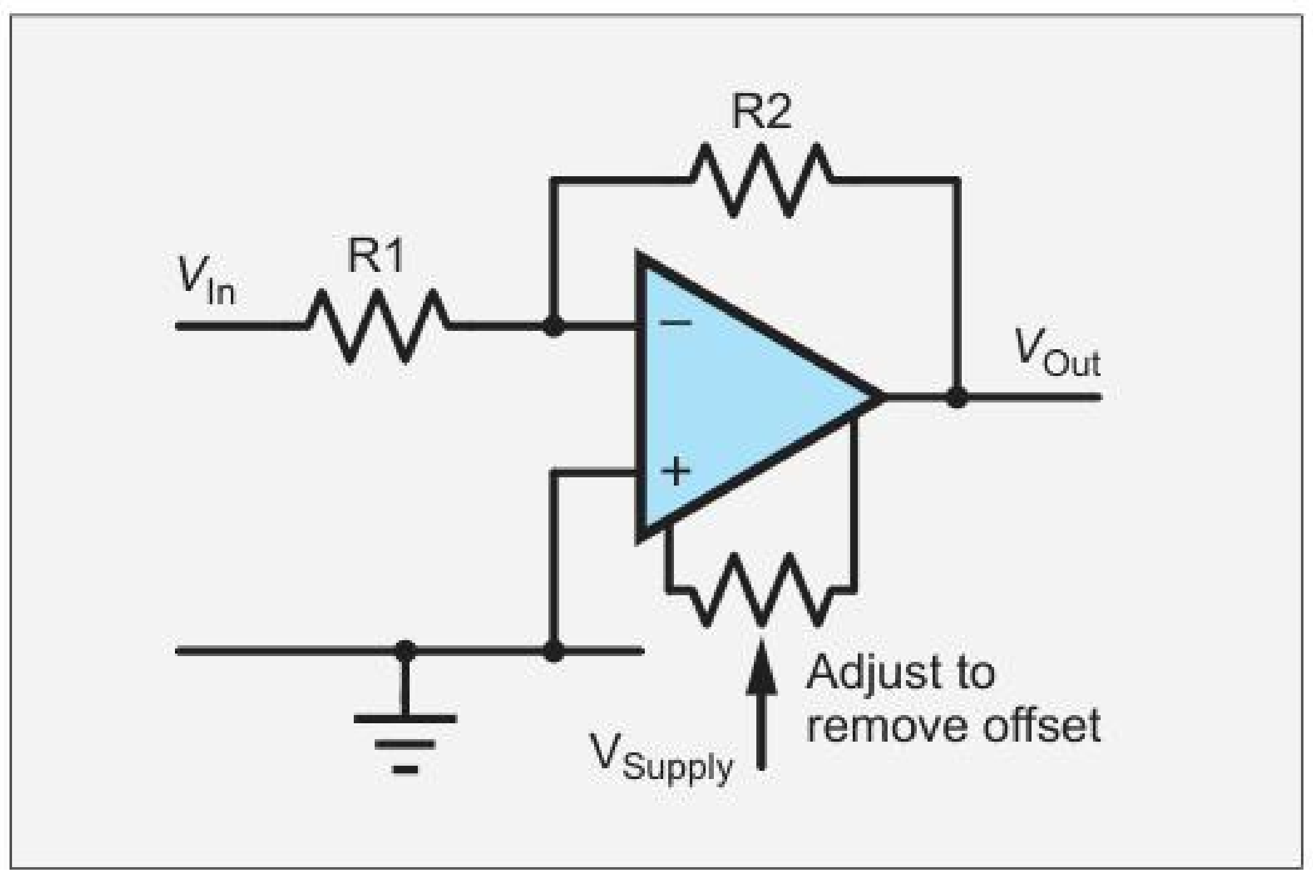


Fig.8. Example offset nulling circuit for an inverting amplifier – check device datasheet for details (eg, which supply the trimmer wiper connects to).

For the OP97 the trimmer can be in the range 5kΩ to 100kΩ and can adjust the offset between 300µV and 850µV depending on the individual device.

Use of nulling pins is often not a good solution for excessive offset. The pins can introduce noise into the circuit if used – particularly with poor layout, such as long leads to the trimmer. Offset drift with temperature and age may mean that the offset has to be nulled more often than just at initial setup, which may be inconvenient. Adding nulling may make temperature drift worse. For these reasons it is often better to find a device with an offset which is inherently small enough for the application. Of course, higher-performance devices may cost more, which may limit options in some cases.

Offset nulling can be achieved without the use of null pins using circuits such as the one in Fig.9. This creates a small, adjustable voltage at the non-inverting input of the op amp in an inverting amplifier. The V_R voltages can be the supply and R_P can be equal to the parallel value of R1 and R2 – which relates to the resistor balancing which we will discuss next month. More details on offset adjustment circuits like this can be found in the *MT-037 Tutorial* document from Analog Devices, see: **https://bit.ly/pe-feb22-ad**

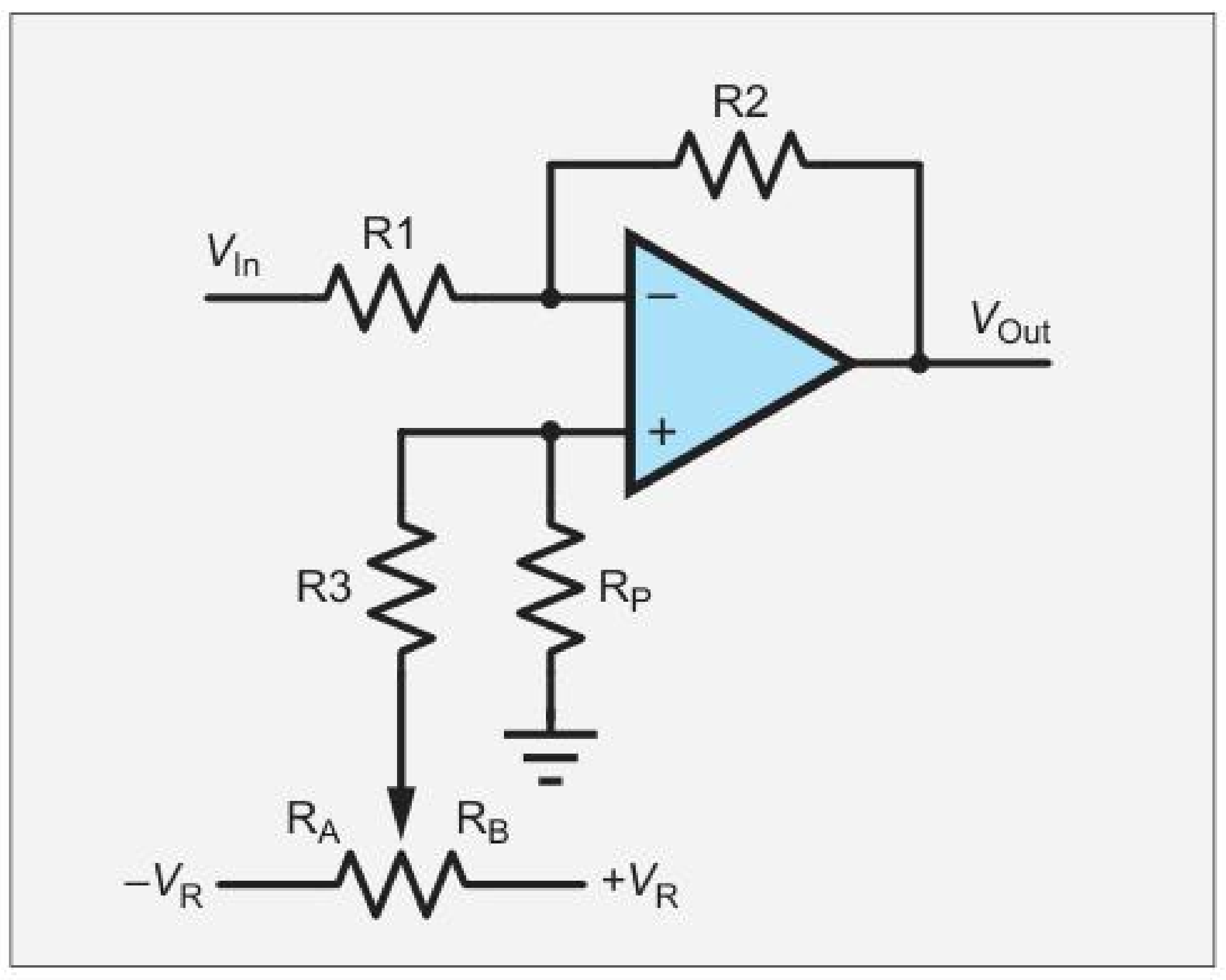


Fig.9. Example of offset nulling circuit using external circuitry rather than op amp null pins.

Simulation files

Most, but not every month, LTSpice is used to support descriptions and analysis in *Circuit Surgery*. The examples and files are available for download from the *PE* website.

STEWART OF READING

17A King Street, Mortimer, near Reading, RG7 3RS
Telephone: **0118 933 1111** Fax: **0118 933 2375**
USED ELECTRONIC TEST EQUIPMENT
Check website **www.stewart-of-reading.co.uk**
(ALL PRICES PLUS CARRIAGE & VAT)
Please check availability before ordering or calling in

Fluke/Philips PM3092 Oscilloscope
2+2 Channel 200MHz Delay TB,
Autoset etc – £250

HP 54600B Oscilloscope
Analogue/Digital Dual Trace 100MHz
Only £75, with accessories £125

LAMBDA GENESYS	PSU GEN100-15 100V 15A Boxed As New	£400	Marconi 2305	Modulation Meter	£250
LAMBDA GENESYS	PSU GEN50-30 50V 30A	£400	Marconi 2440	Counter 20GHz	£295
IFR 2025	Signal Generator 9kHz – 2.51GHz Opt 04/11	£900	Marconi 2945/A/B	Communications Test Set Various Options	POA
IFR 2948B	Communication Service Monitor Opts 03/25 Avionics	POA	Marconi 2955	Radio Communications Test Set	£595
IFR 6843	Microwave Systems Analyser 10MHz – 20GHz	POA	Marconi 2955A	Radio Communications Test Set	£725
R&S APN62	Syn Function Generator 1Hz – 260kHz	£295	Marconi 2955B	Radio Communications Test Set	£800
Agilent 8712ET	RF Network Analyser 300kHz – 1300MHz	POA	Marconi 6200	Microwave Test Set	£1,500
HP8903A/B	Audio Analyser	£750 – £950	Marconi 6200A	Microwave Test Set 10MHz – 20GHz	£1,950
HP8757D	Scaler Network Analyser	POA	Marconi 6200B	Microwave Test Set	£2,300
HP3325A	Synthesised Function Generator	£195	Marconi 6960B	Power Meter with 6910 sensor	£295
HP3561A	Dynamic Signal Analyser	£650	Tektronix TDS3052B	Oscilloscope 500MHz 2.5GS/s	£1,250
HP6032A	PSU 0-60V 0-50A 1000W	£750	Tektronix TDS3032	Oscilloscope 300MHz 2.5GS/s	£995
HP6622A	PSU 0-20V 4A Twice or 0-50V 2A Twice	£350	Tektronix TDS3012	Oscilloscope 2 Channel 100MHz 1.25GS/s	£450
HP6624A	PSU 4 Outputs	£400	Tektronix 2430A	Oscilloscope Dual Trace 150MHz 100MS/s	£350
HP6632B	PSU 0-20V 0-5A	£195	Tektronix 2465B	Oscilloscope 4 Channel 400MHz	£600
HP6644A	PSU 0-60V 3.5A	£400	Farnell AP60/50	PSU 0-60V 0-50A 1kW Switch Mode	£300
HP6654A	PSU 0-60V 0-9A	£500	Farnell XA35/2T	PSU 0-35V 0-2A Twice Digital	£75
HP8341A	Synthesised Sweep Generator 10MHz – 20GHz	£2,000	Farnell AP100-90	Power Supply 100V 90A	£900
HP83630A	Synthesised Sweeper 10MHz – 26.5 GHz	POA	Farnell LF1	Sine/Sq Oscillator 10Hz – 1MHz	£45
HP83624A	Synthesised Sweeper 2 – 20GHz	POA	Racal 1991	Counter/Timer 160MHz 9 Digit	£150
HP8484A	Power Sensor 0.01-18GHz 3nW-10µW	£75	Racal 2101	Counter 20GHz LED	£295
HP8560E	Spectrum Analyser Synthesised 30Hz – 2.9GHz	£1,750	Racal 9300	True RMS Millivoltmeter 5Hz – 20MHz etc	£45
HP8563A	Spectrum Analyser Synthesised 9kHz – 22GHz	£2,250	Racal 9300B	As 9300	£75
HP8566B	Spectrum Analsyer 100Hz – 22GHz	£1,200	Solartron 7150/PLUS	6½ Digit DMM True RMS IEEE	£65/£75
HP8662A	RF Generator 10kHz – 1280MHz	£750	Solatron 1253	Gain Phase Analyser 1mHz – 20kHz	£600
Marconi 2022E	Synthesised AM/FM Signal Generator 10kHz – 1.01GHz	£325	Solartron SI 1255	HF Frequency Response Analyser	POA
Marconi 2024	Synthesised Signal Generator 9kHz – 2.4GHz	£800	Tasakago TM035-2	PSU 0-35V 0-2A 2 Meters	£30
Marconi 2030	Synthesised Signal Generator 10kHz – 1.35GHz	£750	Thurlby PL320QMD	PSU 0-30V 0-2A Twice	£160 – £200
Marconi 2023A	Signal Generator 9kHz – 1.2GHz	£700	Thurlby TG210	Function Generator 0.002-2MHz TTL etc Kenwood Badged	£65

HP33120A	Function Generator 100 microHz – 15MHz	£350
HP53131A	Universal Counter 3GHz Boxed unused	£600
HP53131A	Universal Counter 225MHz	£350
Audio Precision	SYS2712 Audio Analyser – in original box	POA
Datron 4708	Autocal Multifunction Standard	POA
Druck DPI 515	Pressure Calibrator/Controller	£400
Datron 1081	Autocal Standards Multimeter	POA
ENI 325LA	RF Power Amplifier 250kHz – 150MHz 25W 50dB	POA
Keithley 228	Voltage/Current Source	POA
Time 9818	DC Current & Voltage Calibrator	POA

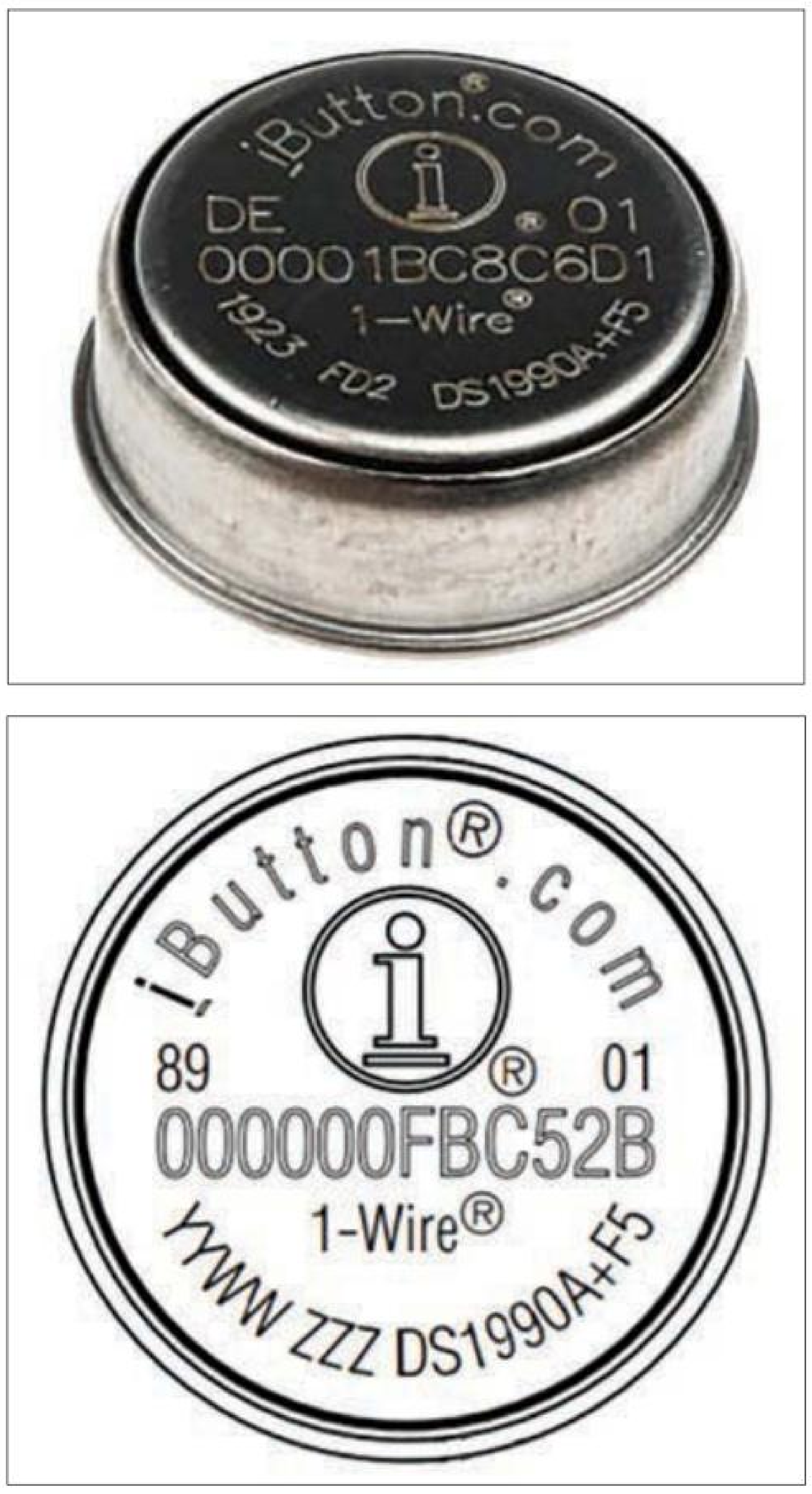
HP/Agilent HP 34401A Digital
Multimeter 6½ Digit £325 – £375

Marconi 2955B Radio
Communications Test Set – £800

Make it with Micromite

Phil Boyce – hands on with the mighty PIC-powered, BASIC microcontroller

Part 33: Using CRC with iButtons and the Micromite



A reminder from last month of what iButtons look like. Note in the bottom diagram the location of the family code byte (01) and the CRC (89).

Last month, we explored how to use a low-cost iButton (the DS1990) with a Micromite. We discussed the use of an iButton reader (or a pair of wires) to electrically connect an iButton to a single Micromite I/O pin (and ground). By using MMBASIC's built-in 1-wire commands, we then showed how easy it is to detect the presence of an iButton.

Micromite code

The code in this article is available for download from the *PE* website.

We explained that different types of iButton are available, which all have a unique ID number which is 64-bits (8-bytes) long. We highlighted that one of those eight bytes represents the type of iButton (ie, the Family Code), and another byte represents what is referred to as the CRC (cyclic redundancy check). The **iButtonDemo2.txt** program from last month reads the eight bytes from the iButton (essentially reading 64-bits of serial data on the 1-wire bus) and then displays the result as eight hexadecimal (hex) byte values. These eight bytes form the unique ID number, and they should be an exact match to the 8-byte ID number that is engraved (in hex) onto the iButton's case. However, any contact-bounce during the serial reading of the 64-bits will result in corrupt data values; so how can we be 100% sure that the ID number read is correct?

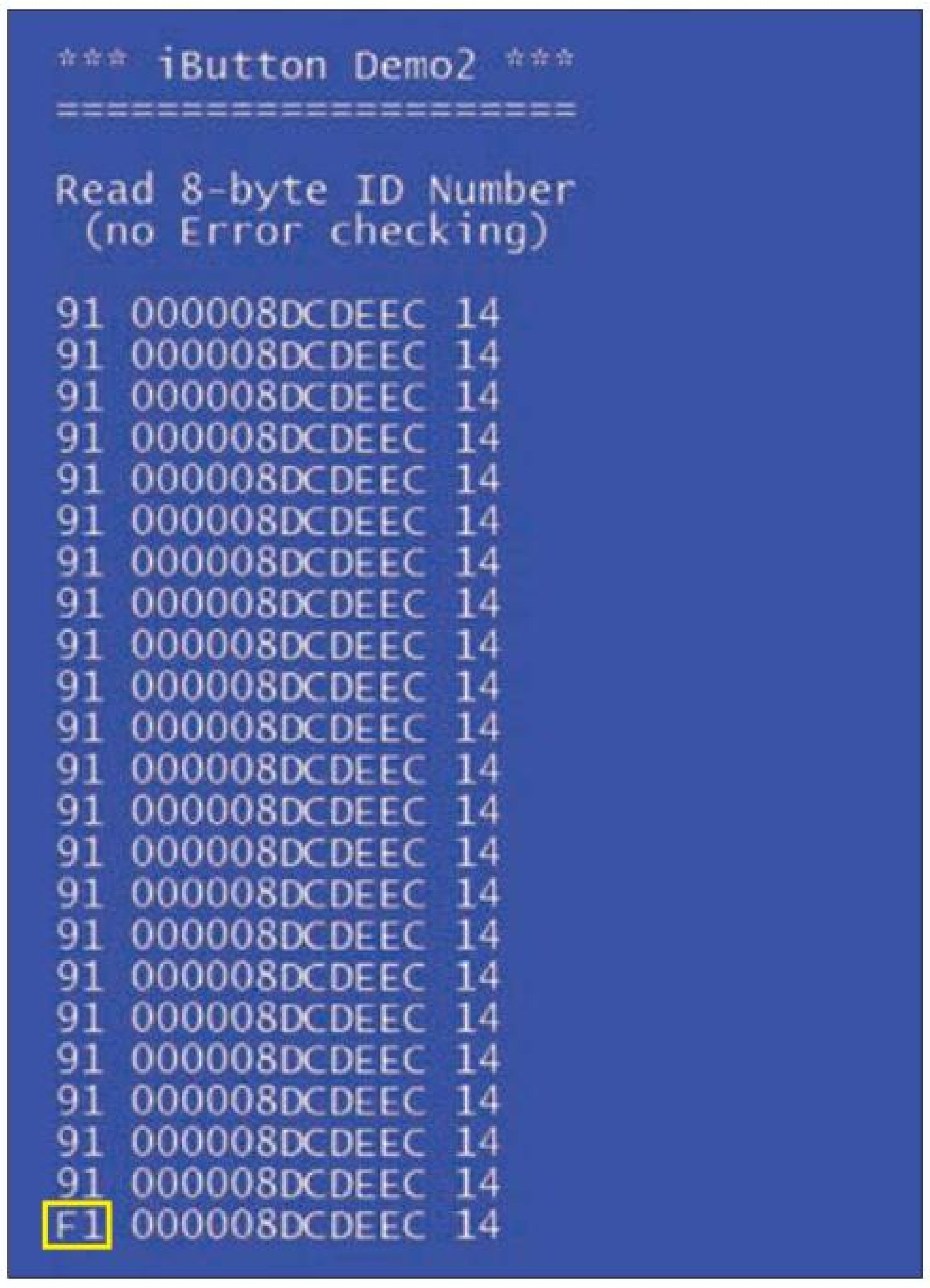
This month, we will begin by explaining how the **iButtonDemo2.txt** program extracts the 8-byte ID value, and then go into detail as to how to use the CRC byte to check whether the data read is correct and valid (ie, no contact bounce during the reading process), or whether it is corrupted and hence should be ignored. This is an important topic to understand before introducing a different type of iButton, the DS1971, which incorporates 256 bits of EEPROM. This specific EEPROM iButton allows you to write up to 32 bytes of custom data to it – for example, a person's name. This iButton will be explored next time and in preparation for our *Electronic Combination Door Lock* project, in which EEPROM iButtons will be used as the 'keys', and a touchscreen display will be used to show the name of the person attempting to unlock the door (the name being stored in EEPROM).

Let's get started by explaining more about last month's downloadable program.

iButtonDemo2.txt

When you `RUN` **iButtonDemo2.txt** and connect an iButton to the Micromite, you should see that eight hex bytes are displayed on the Terminal screen (repeatedly while the iButton is present). What you may notice is that for most of the time, the eight hex byte values displayed (on each line) are identical, line-by-line; and they are also the same as the eight hex values that are stamped onto the iButton's case. However, occasionally you may see a line of different values displayed, and this is simply due to contact bounce (see Fig.1) – more on this shortly.

Communication starts with the Micromite (the 'master') sending a command (ie, a series of pulses) on the

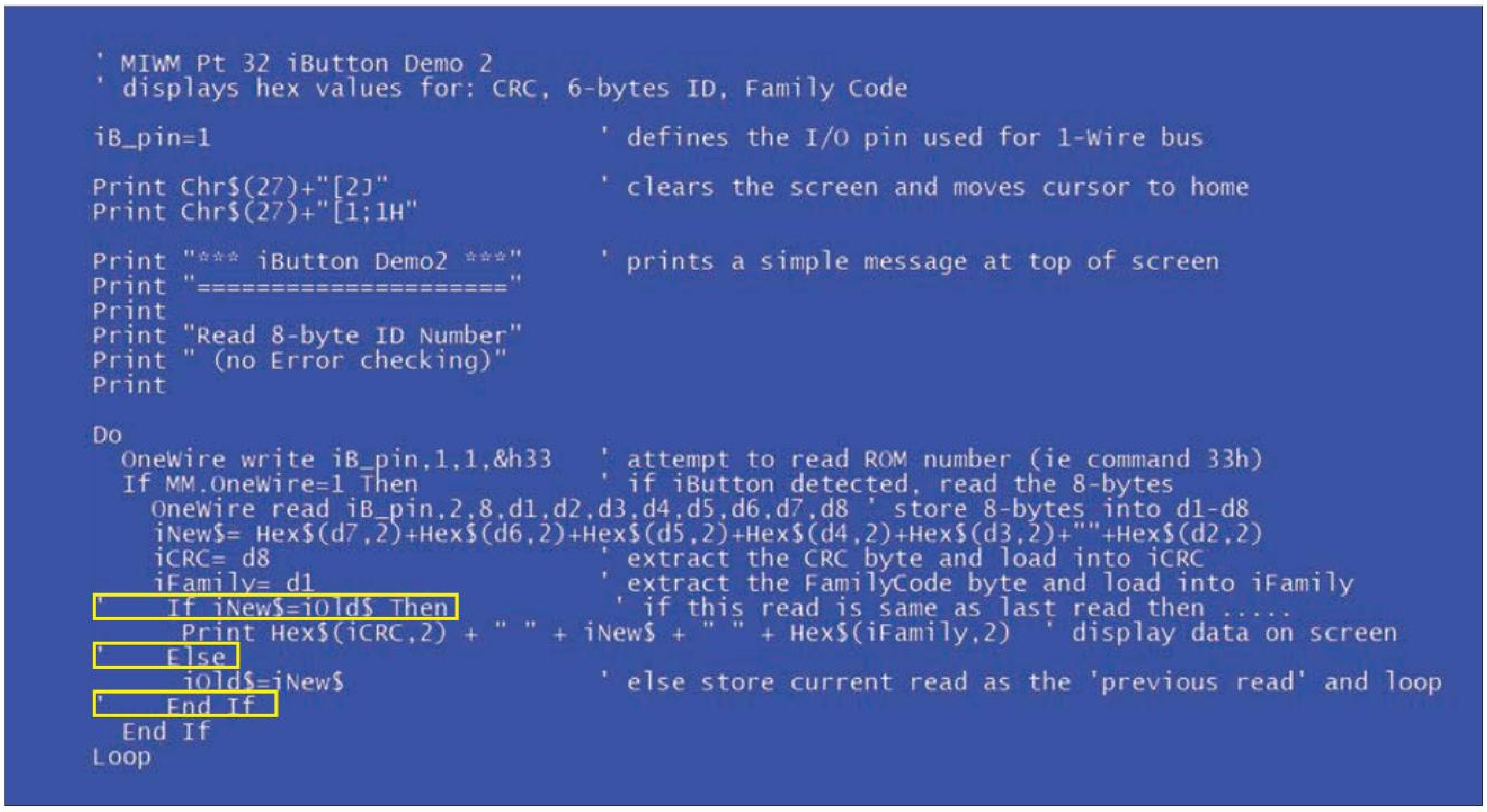


```
*** iButton Demo2 ***
=====================

Read 8-byte ID Number
 (no Error checking)

91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
F1 000008DCDEEC 14
```

Fig.1. An example of the output from **iButtonDemo2.txt** program while an iButton is connected to the Micromite. Note the last line of data shows the impact of contact bounce as the iButton was removed.



```
' MIWM Pt 32 iButton Demo 2
' displays hex values for: CRC, 6-bytes ID, Family Code

iB_pin=1                           ' defines the I/O pin used for 1-Wire bus

Print Chr$(27)+"[2J"               ' clears the screen and moves cursor to home
Print Chr$(27)+"[1;1H"

Print "*** iButton Demo2 ***"      ' prints a simple message at top of screen
Print "====================="
Print
Print "Read 8-byte ID Number"
Print " (no Error checking)"
Print

Do
  OneWire write iB_pin,1,1,&h33    ' attempt to read ROM number (ie command 33h)
  If MM.OneWire=1 Then             ' if iButton detected, read the 8-bytes
    OneWire read iB_pin,2,8,d1,d2,d3,d4,d5,d6,d7,d8 ' store 8-bytes into d1-d8
    iNew$= Hex$(d7,2)+Hex$(d6,2)+Hex$(d5,2)+Hex$(d4,2)+Hex$(d3,2)+""+Hex$(d2,2)
    iCRC= d8                       ' extract the CRC byte and load into iCRC
    iFamily= d1                    ' extract the FamilyCode byte and load into iFamily
'    If iNew$=iOld$ Then            ' if this read is same as last read then .....
      Print Hex$(iCRC,2) + " " + iNew$ + " " + Hex$(iFamily,2)  ' display data on screen
'    Else
      iOld$=iNew$                  ' else store current read as the 'previous read' and loop
'    End If
  End If
Loop
```

Fig.2. Comment out these three lines of code from the **iButtonDemo2.txt** program to better demonstrate contact bounce.

1-wire bus. In our program, this is the Read ROM command:

```
ONEWIRE WRITE iB_pin,1,1,&h33
```

If an iButton is present, then the initial pulses provide parasitic power (see last month) which powers the iButton's internal circuit long enough for it to respond to the command it was sent. This response is a series of pulses that represent the data that needs to be sent back to the Micromite, or to put it another way, the iButton toggles the logic level on the 1-wire bus in a controlled manner so that the MMBASIC firmware can sample the logic level on the 1-wire bus to receive the data being sent by the iButton. Clearly this all needs to be very precisely timed, and thankfully, the MMBASIC firmware deals with all of this for us, thereby making iButton communication easy. Note that when MMBASIC sees a response on the 1-wire bus, it sets the system variable `MM.ONEWIRE` to a value of `1`. However, if there is no response detected on the 1-wire bus, then it is assumed that no iButton is present, and MMBASIC sets the system variable `MM.ONEWIRE` to a value of `0`.

The value of the `MM.ONEWIRE` system variable is what we used in **Demo1** when demonstrating how to detect the presence of an iButton. In **Demo2**, it is used with an `IF…THEN…` command, which in turn, is inside a `DO/LOOP` to continually look for the presence of an iButton on the 1-wire bus. Whenever an iButton is present, the block of code inside the `IF/ENDIF` section will read the iButton response (eight bytes) and load the values into eight variables, `d1` to `d8`. This data read is performed with the single line of code:

```
ONEWIRE READ
iB_pin,2,8,d1,d2,d3,d4,d5,d6,d7,d8
```

Once these variables are loaded, they are re-sequenced, and then converted into hex so that the layout matches the eight hex bytes that are stamped on the iButton's case. To clarify, `d8` contains the CRC byte and is displayed on the left (hex value `91` in Fig.1) followed by the next 6-bytes of the ID number (`d7` to `d2`). These six bytes are formatted and loaded into the string, `iNew$`. Finally, variable `d1` is displayed on the right and this contains the Family Code byte (hex value `14` in Fig.1, which represents a DS1971 EEPROM iButton).

The line: `IF iNew$=iOld$ THEN` is there to ensure that the eight hex bytes are only displayed when the six middle bytes read 'this time' match the six middle bytes read 'last time'. This will remove a lot of contact bounce as we will demonstrate shortly. Note that any difference between successive reads with the first byte displayed (`d8`, CRC) or the last byte displayed (`d1`, Family Code) is not filtered out (as can be seen in Fig.1).

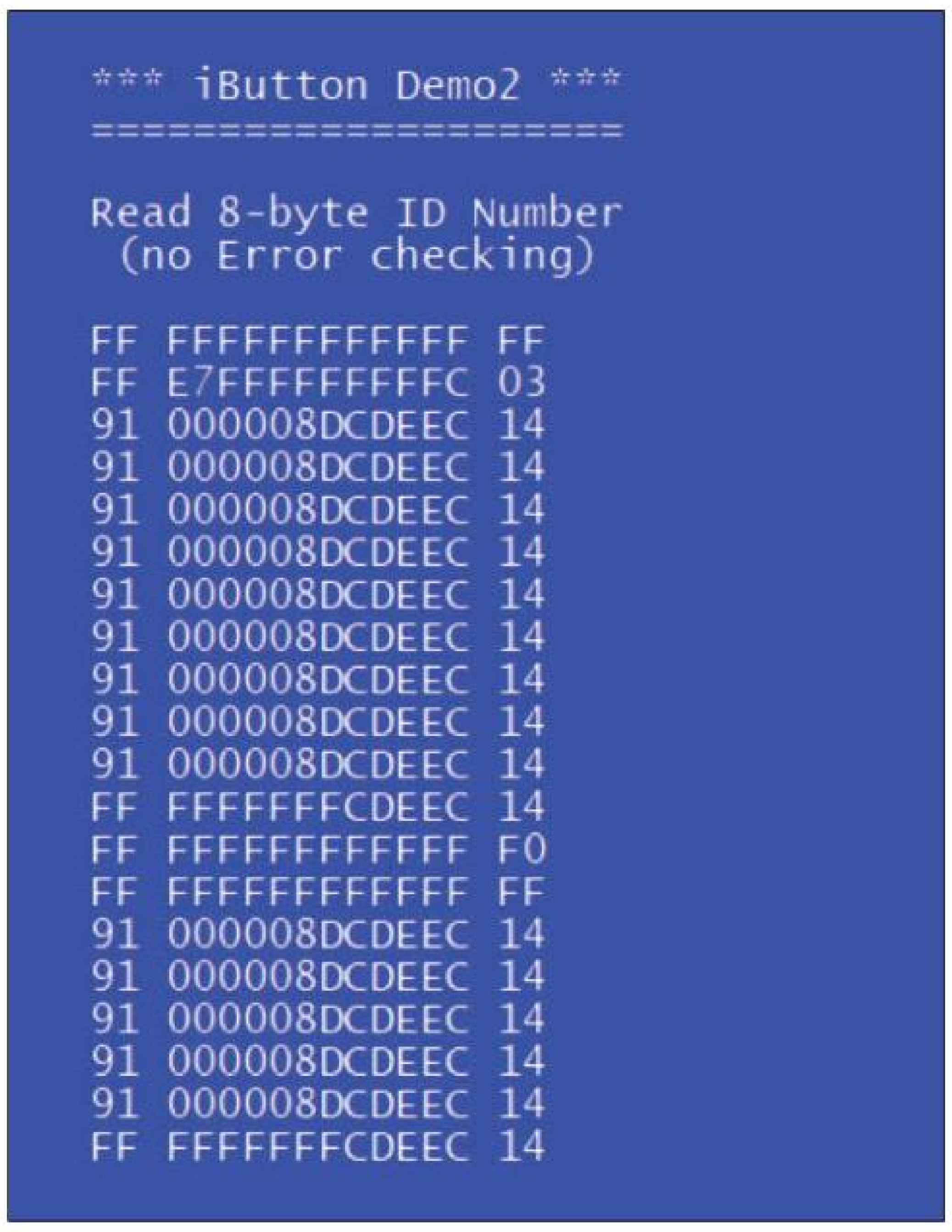
Contact bounce

To better demonstrate contact bounce, `EDIT` the code and comment out the three lines associated with the `IF`/`ELSE`/`ENDIF` near the end of the `DO/LOOP` (refer to Fig.2). Now `RUN` the program and observe what happens when you present an iButton; you should see a lot more contact bounce in the form of different line-by-line values (see Fig.3).

What you will probably notice is that there are a lot more `FF` hex values being displayed; we will now explain the reason for this.

If you refer to the schematic (see Fig.8 from last month), you will see the 1-wire bus is pulled high by a pull-up resistor. When the iButton needs to transmit a `0` data bit, all it needs to do is to simply pull the 1-wire bus to ground (and the Micromite will then sample a low logic level representing the value of `0`). And when the iButton needs to transmit a `1` data bit, it doesn't need to do anything at all since the pull-up resistor will ensure that the Micromite samples a high logic level (representing the value of `1`).

This will all work correctly when the iButton is electrically connected to the Micromite; however, in reality (and at the macroscopic level) there will be a lot of contact bounce that occurs. What this means is that the iButton is rapidly altering between being in contact with the Micromite, and not in contact at all. When it is not in contact (during the 'bouncing'), the Micromite will sample as normal, but will only see a logic high due to the pull-up resistor. If this occurs across all eight sampling timeslots (ie, bits) that form a byte value, then the result seen by the



```
*** iButton Demo2 ***
=====================

Read 8-byte ID Number
 (no Error checking)

FF FFFFFFFFFFFF FF
FF E7FFFFFFFFFC 03
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
FF FFFFFFFCDEEC 14
FF FFFFFFFFFFFF F0
FF FFFFFFFFFFFF FF
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
91 000008DCDEEC 14
FF FFFFFFFCDEEC 14
```

Fig.3. An example output from the modified **iButtonDemo2.txt** program to better demonstrate contact bounce.

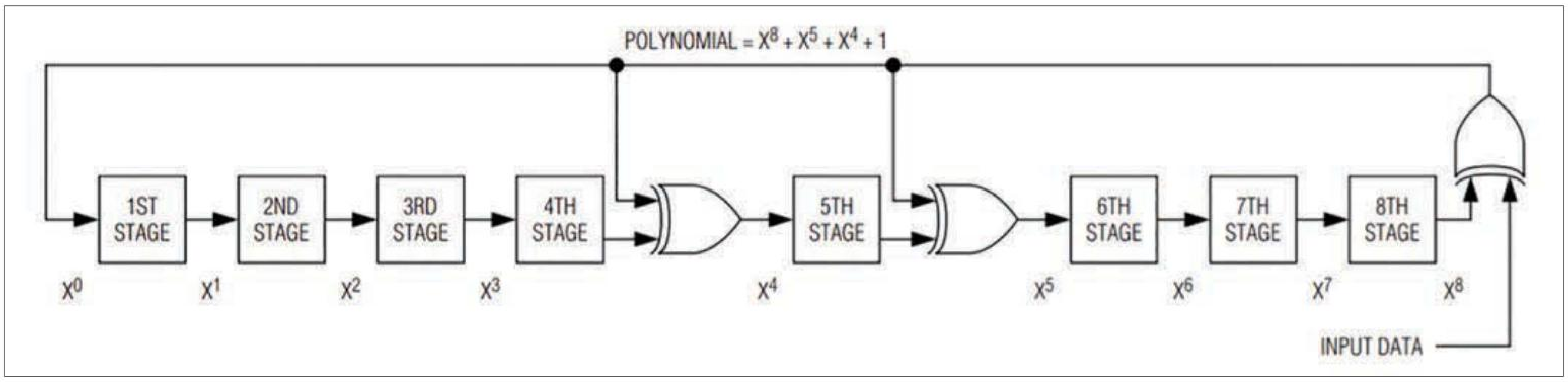


Fig.4. The 1-wire CRC generator in schematic form, as taken from Maxim's DS1990 datasheet.

Micromite is going to be a binary value of `11111111`, or in other words, `FF` hex.

Hopefully this has explained the issue of contact bounce, and has also highlighted its effect on the data read by the Micromite. What we now need to understand is how to eliminate rogue data received by the Micromite so that we can be 100% certain that the 8-bytes received are in fact a true (and correct) representation of the iButton's actual ID number, as stamped on its casing. This is where the CRC byte comes into play.

CRC

As already mentioned, the CRC byte is part of the iButton's 8-byte ID number, and it can be used to check the validity of the ID number received by the Micromite. The simplest way to consider its purpose is that when the other seven-byte values received are passed through a specific algorithm (defined by the original manufacturer, Dallas Semiconductors) the output (ie, answer) should be the exact same value as the CRC byte. If it is, then the 8-byte ID value received can be considered as being 100% correct. However, if the answer is different, then the data received should be disregarded (and considered corrupt, most likely due to contact bounce).

So how exactly does the CRC checking work? Unfortunately, the mathematical theory behind it all is far too complex to go into here, but the steps needed to be performed to simulate the algorithm can be explained relatively easily. For those interested in the complexity behind it, an online application note can be found at: **https://bit.ly/pe-feb22-max**

Also, if you examine an iButton's datasheet, you will see reference to the CRC, and mention that the maths is based on the polynomial $x^8 + x^5 + x^4 + 1$, which in turn can be represented by a schematic diagram – see Fig.4 (taken from the DS1990 datasheet). The reason the schematic is recreated here is so that you can see some connection with the following explanation of the steps taken by the algorithm.

The CRC process

The complete CRC process will effectively take the first seven bytes received (ie, `d1` to `d7` as used in **Demo2**), and along with the specific rules of the algorithm, each of these bytes will be used in turn to adjust the value of a variable that we will call `CalcCRC`. By the time we have used the last bit of the seventh byte (`d7`) to adjust `CalcCRC`, the value of `CalcCRC` should be the same as the CRC byte value (`d8`). Furthermore, and for completeness, if `d8` is also used to adjust `CalcCRC` (but only when `CalcCRC` has the same value as `d8`; as it should if everything is received correctly), then the final value of `CalcCRC` will be `0`. However, there is no need to do this extra step as we can simply stop at `d7` and check that it has the same value as `d8`.

We need to begin the whole process by setting `CalcCRC` to an initial value of `0`, and then to apply the algorithm, we need to work through each of the 8-bits (least significant bit (LSB) first) of the first byte received (`d1`) and use each bit to determine how the value of `CalcCRC` is adjusted (more on exactly what happens will be explained shortly). This process is then repeated, with the remaining bytes (`d2` to `d7`), with each byte continually adjusting the value of `CalcCRC`. In simple terms, we need to adjust the value of `CalcCRC` (which starts at an initial value of `0`) a total of 56 times (7-bytes, 8-bits per byte) to arrive at the final answer that should match the CRC value.

The CRC algorithm

This is where it starts to get a little trickier, but please do be patient, take your time, and try to follow things since it will prove extremely useful to be able to understand how things work here – not to mention, it is very satisfying once understood!

Referring to Fig.4, consider the bits representing the current value of `CalcCRC` being put into the eight boxes, (MSB in the 1st stage, and LSB in the in the 8th stage). The bit-value of the 8th stage (ie, the LSB of `CalcCRC`) needs to be XOR-ed with the input data bit – let us call the outcome the 'XOR result'. Note that the input data bit comes from the current bit value from the data byte value being passed (ie, from `d1` to `d7`). If the `XOR` result is a `0`, then the 8-bits of `CalcCRC` are simply shifted one place to the right, resulting in a new value for `CalcCRC`. However, if the XOR result is a `1`, then not only are the bits of `CalcCRC` shifted to the right, but in addition, three bits of `CalcCRC` (the bits in stages `1`, `5`, and `6`) are also XOR-ed with `1`. If you work through the combinations of the XOR truth table, you will see that this simply means inverting these three bits.

To help reinforce the above explanation, let us work through two different scenarios – one where just a simple shift of `CalcCRC` is required to create the new value of `CalcCRC`, and the other where we also need to invert the relevant bits of `CalcCRC` to generate its new value.

CRC example

For this example, let us assume that we have already passed all 48-bits from `d1` to d6, through the algorithm, along with the first 6-bits from `d7` (ie, bit 0 to bit 5), and hence we now only need to apply the last two bits (bit 6 and bit 7) from `d7` into the algorithm – in other words, two more `CalcCRC` values to generate before comparing its final value with that of `d8`.

Let's assume that the current value for `CalcCRC` is hex `76` (in binary: `01110110`) and let us also use a value of hex `00` for `d7` (in binary `00000000`).

So, the LSB (bit-0) from `CalcCRC` (0111011**0**) needs to be XOR-ed with the input data bit which now comes from bit 6 of `d7` (0**0**000000). The result of `0 XOR 0` is `0` which means we just need to shift the `CalcCRC` bits (currently `01110110`) one place to the right to generate its new value of: `00111011`.

Now let's do the final pass by using the new value of `CalcCRC` generated above (`00111011`), and bit 7 of `d7`. Hence, the LSB (bit 0) from `CalcCRC` (now 0011101**1**) needs to be XOR-ed with the input data bit which now comes from bit 7 of `d7` (**0**0000000). The result of `1 XOR 0` is `1`, which means we not only need to shift the `CalcCRC` bits (currently `00111011`) to the right to generate its new value of: **0**001**11**01, but we also need to invert the relevant bits (shown in bold and underlined) to generate the new value of: **1**001**00**01.

So, the final value of `CalcCRC` is `10010001` which is hex `91`. This can now be compared to `d8`, and if it matches, we can assume the 8-byte ID number read from the iButton is correct.

In the example just shown, we defined the current value of `CalcCRC` as hex `76`. This was not a value chosen at random, but was chosen intentionally as it is the actual value that would have been reached for `CalcCRC` after passing the first 54 bits into the algorithm (and with everything operating correctly) when using the following values:

```
d1='14'
d2='EC'
d3='DE'
d4='DC'
d5='08'
d6='00'
d7='00'
```

So why use these values? The observant among you will see that these are the exact same values as in the screenshot from Fig.1 (or Fig.3). Remember that the CRC byte (`d8`) is displayed as the left-hand byte in each row, and is indeed showing a hex value of `91`, and hence here we can be assured that the variables `d1` to `d8` all contain correctly read data values. Furthermore, `d1` (the Family Code byte value shown on the right of each row) is showing a hex value of `14`, and this represents a DS1971 256-bit EEPROM iButton (the iButton we will be using later).

iButtonDemo3.txt

The above example of performing the final two passes of the algorithm should help you understand things much better (if not, take a break, and read it again another time!).

For an extreme challenge, why not start at the beginning, and work through all 56 passes of the algorithm using the above values for `d1` to `d7` (and using `0` as the initial `CalcCRC` value, as would be required). This would clearly be a lengthy task to perform manually, and therefore we have created a download for you (**iButtonDemo3.txt**) which shows the new value of `CalcCRC` after each stage (see Fig.5). In fact, it also passes `d8`, showing that the final result of `CalcCRC` is indeed `0` (but only when `d1` to `d8` contain correct values). Download the program and note that the code is commented throughout, but don't worry if you don't understand fully how it works.

There are two lines of code in the program that need highlighting to understand what is going on:

1. `CalcCRC=CalcCRC>>1`

This line simply shifts the bits in the variable `CalcCRC` one place to the right as is the requirement in the CRC algorithm. Note that what was contained in bit 0 (the right-hand bit) is simply discarded, and the new MSB (ie, bit 7 in our case) is set to `0`. As an example, if `CalcCRC` contains the binary value `10111101`, then the result of the above `>>1` would be `01011110`

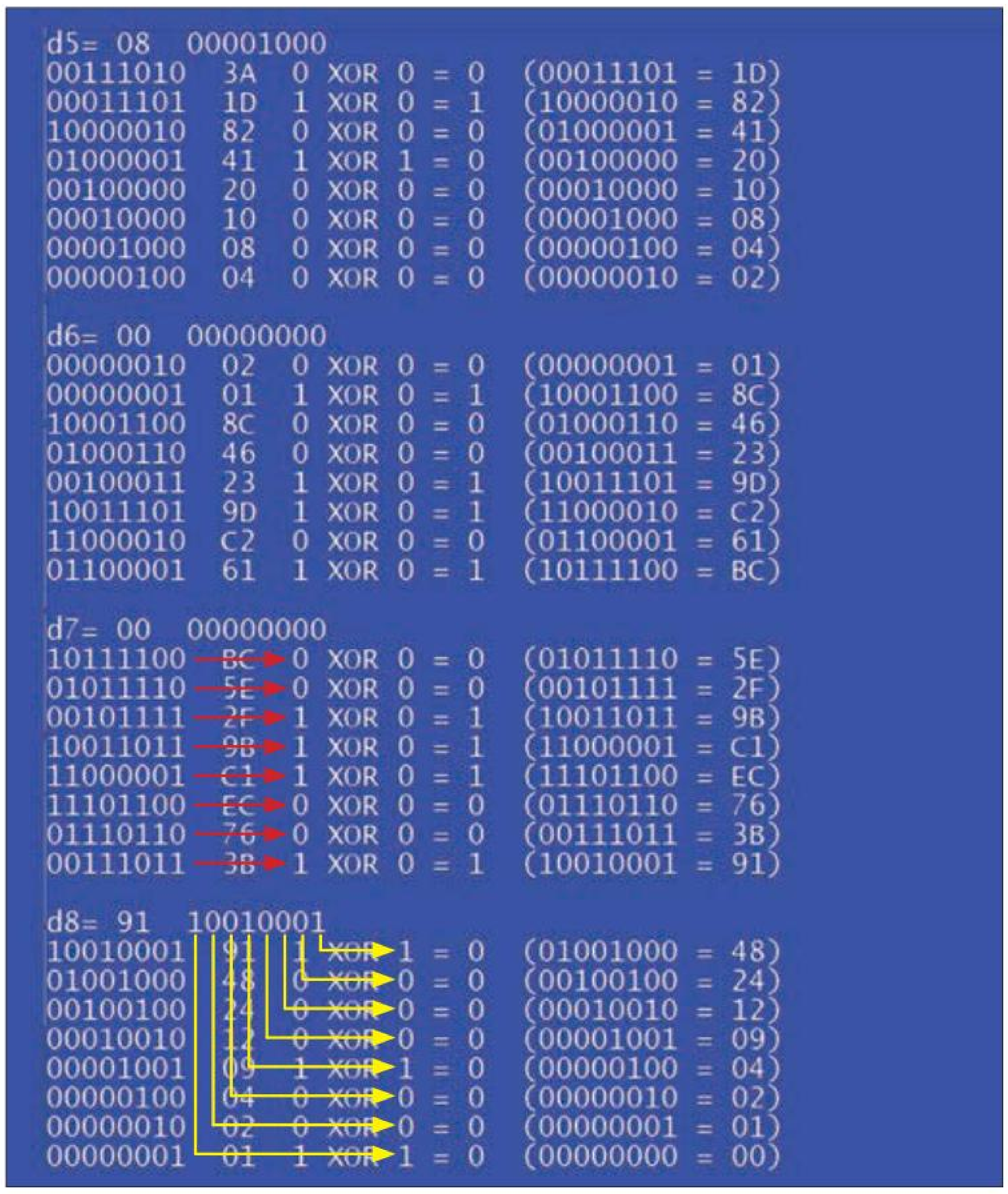
2. `CalcCRC=CalcCRC XOR &h8C`

This line has the effect of inverting bit 7, bit 3, and bit 2, which sometimes is a requirement in the CRC algorithm. The hex value `8C` (in the above line of code) equates to `10001100` in binary. So, if `CalcCRC` contains the binary value `11111111`, then the above becomes:

1111**11**11 XOR 10001100 equals **0**111**00**11

Which has only inverted the bits highlighted in bold.

Likewise, if `CalcCRC` contains the binary value `00000000`, then the above becomes:



```
d5= 08   00001000
00111010  3A  0 XOR 0 = 0  (00011101 = 1D)
00011101  1D  1 XOR 0 = 1  (10000010 = 82)
10000010  82  0 XOR 0 = 0  (01000001 = 41)
01000001  41  1 XOR 1 = 0  (00100000 = 20)
00100000  20  0 XOR 0 = 0  (00010000 = 10)
00010000  10  0 XOR 0 = 0  (00001000 = 08)
00001000  08  0 XOR 0 = 0  (00000100 = 04)
00000100  04  0 XOR 0 = 0  (00000010 = 02)

d6= 00   00000000
00000010  02  0 XOR 0 = 0  (00000001 = 01)
00000001  01  1 XOR 0 = 1  (10001100 = 8C)
10001100  8C  0 XOR 0 = 0  (01000110 = 46)
01000110  46  0 XOR 0 = 0  (00100011 = 23)
00100011  23  1 XOR 0 = 1  (10011101 = 9D)
10011101  9D  1 XOR 0 = 1  (11000010 = C2)
11000010  C2  0 XOR 0 = 0  (01100001 = 61)
01100001  61  1 XOR 0 = 1  (10111100 = BC)

d7= 00   00000000
10111100  BC  0 XOR 0 = 0  (01011110 = 5E)
01011110  5E  0 XOR 0 = 0  (00101111 = 2F)
00101111  2F  1 XOR 0 = 1  (10011011 = 9B)
10011011  9B  1 XOR 0 = 1  (11000001 = C1)
11000001  C1  1 XOR 0 = 1  (11101100 = EC)
11101100  EC  0 XOR 0 = 0  (01110110 = 76)
01110110  76  0 XOR 0 = 0  (00111011 = 3B)
00111011  3B  1 XOR 0 = 1  (10010001 = 91)

d8= 91   10010001
10010001  91  1 XOR 1 = 0  (01001000 = 48)
01001000  48  0 XOR 0 = 0  (00100100 = 24)
00100100  24  0 XOR 0 = 0  (00010010 = 12)
00010010  12  0 XOR 0 = 0  (00001001 = 09)
00001001  09  1 XOR 1 = 0  (00000100 = 04)
00000100  04  0 XOR 0 = 0  (00000010 = 02)
00000010  02  0 XOR 0 = 0  (00000001 = 01)
00000001  01  1 XOR 1 = 0  (00000000 = 00)
```

Fig.5. The **iButtonDemo3.txt** program displays the new value of `CalcCRC` after each pass of the CRC algorithm.

0000**00**00 XOR 10001100 equals **1**000**11**00

Which, again, has only inverted the bits highlighted in bold.

The reason for going into the detail of the CRC checking is that it is essential to ensure that any serial data read from an iButton is 100% correct (ie, free of contact bounce). With the DS1990 ID-only button, we only have the 8-byte ID number to concern ourselves with; however, we will be using the DS1971 EEPROM iButton for use in our upcoming *Electronic Combination Door Lock* project, in which we will also need to read 32 bytes of custom data (containing a person's name). You can see that when reading 32 bytes there is much more chance of contact bounce during the serial read compared to reading just eight bytes; hence, CRC checking is a critical part of the solution.

There are 32-bytes available in the DS1971, so the concept will be to use the first 31 bytes to store the person's name, and then use the 32nd byte to store the CRC value (much like the last byte in the 8-byte ID number for any iButton is the CRC byte programmed in by the manufacturer).

The obvious question is how do we generate the CRC value to be stored in byte-32 of the EEPROM iButton? Helpfully, the CRC algorithm can be used to generate a CRC byte value after passing it *any* number of bytes; so we can therefore pass 31-bytes into the algorithm, and generate the final value of `CalcCRC` which can then be written as byte 32 into the EEPROM.

Next time

Having got a grasp of dealing with contact bounce, next month we will explore the DS1971 256-bit EEPROM iButton and demonstrate how to write 32-bytes of custom data into EEPROM (including the CRC byte value that we need to generate), along with how to read this data back into the Micromite – error free.

Until then, stay safe, and have FUN!

Questions? Please email Phil at:
contactus@micromite.org

Practical Electronics PCB SERVICE

PROJECT	CODE	PRICE
FEBRUARY 2022		
Arduino-based Power Supply	18106201	9.95
Battery Monitor Logger	11106201	10.95
Electronic Wind Chimes	23011201	10.95
Analogue Vocoder – Driver Amplifier	AO-FEB22	8.95
JANUARY 2022		
Vintage battery Radio Li-ion Power Supply	11111201	9.95
MiniHeart: A Miniature Heartbeat Simulator	01109201	8.95
DECEMBER 2021		
AM/FM/SW Digital Receiver	CSE200902A	13.95
Balanced Input and Attenuator for USB CODEC	01106202	11.95
NOVEMBER 2021		
Dual Battery Lifesaver	11111202	£6.95
OCTOBER 2021		
Mini Wi-Fi LCD BackPack	24106201	£8.95
SEPTEMBER 2021		
USB SuperCodec PCB	01106201	£14.95
Audio DDS Oscillator PCB	01110201	£5.95
Audio DDS Oscillator rotary encoder	01110201-ENC	6.95
Programming Adaptor Board for Audio DDS Oscillator	01110202	£5.95
High-power Ultrasonic Cleaner main PCB	04105201	£14.95
High-power Ultrasonic Cleaner front-panel PCB	04105202	
Night Keeper Lighthouse PCB	08110201	£5.95
AUGUST 2021		
Ol' Timer PCB	19104201	£11.95
Ol' Timer 8x8 RGB LED module using WS2812B	19104201-88	£8.95
Ol' Timer set of acrylic case pieces and spacer	19104201-ACR	£8.75
Ol' Timer DS3231 RTC IC wide SOIC-16	19104201-RTC	£5.95
Wideband Digital RF Power Meter	04106201	£9.75
Switchmode 78xx regulators ***(PACK of 5!)***	18105201	£7.95
Cool Beans SMAD display	CB-AUG21	£11.95
JULY 2021		
ATtiny816 Breakout / Dev Board with Capacitive Touch	24110181	£9.75
IR Remote Control Assistant (Jaycar version)	15005201	£8.95
IR Remote Control Assistant (Altronics version)	15005202	£8.95
PIC18F Development Board	PNM-JUL21	£12.95
Microphone Preamplifier	AO-JUL21	£11.95
JUNE 2021		
Roadies' Test Signal Generator (surface-mount version)	01005201	£8.95
Roadies' Test Signal Generator (through-hole version)	01005202	£9.95
Touchscreen Wide-range RCL Box (Resistor module)	04104201	£18.95
Touchscreen Wide-range RCL Box (Ind/Cap module)	04104202	
KickStart Part 3 – Gyrator-based Audio Filter	KS3-2021	£7.95
MAY 2021		
7-Band Equaliser (Mono)	01104201	£8.95
7-Band Equaliser (Stereo)	01104202	£10.95
Car Altimeter	05105201	£7.95
APRIL 2021		
Reflow Oven – DSP Active Crossover (CPU)	01106193	£19.95
Reflow Oven – DSP Active Crossover (Front panel)	01106195	
Reflow Oven – DSP Active Crossover (LCD)	01106196	
Frequency Reference Signal Distributor	CSE200103	£8.95
MARCH 2021		
Nutube Guitar Effects Pedal	01102201	£12.95
Programmable Thermal Regulator (Peltier Interface)	21109181	£18.95
Programmable Thermal Regulator (Peltier Driver)	21109182	
Tunable HF Preamp	CSE190502	£8.95
FEBRUARY 2021		
4G Remote Monitoring	27111191	£9.95
JANUARY 2021		
Nutube Valve Preamplifier	01112191	£12.95
Arduino DCC Controller	09207181	£10.95
DECEMBER 2020		
Pseudo-Random Sequence Generator	16106191	£7.95
Clever Charger	14107191	£11.95
PE Theremin Amplifier	AO-1220-01	£8.95
NOVEMBER 2020		
LED Christmas Tree (1 off)	16107181-1	£6.95
LED Christmas Tree (4 off)	16107181-2	£14.95
LED Christmas Tree (12 off)	16107181-3	£24.95
LED Christmas Tree (20 off)	16107181-4	£34.95
USB/SPI Interface Board	16107182	£5.95
45V/8A Power Supply PCB *plus* acrylic spacer	18111181	£14.95
45V/8A Power Supply front panel five-way display bezel	18111181-BZ	£3.95
Five-way LCD Panel Meter/Display	18111182	£7.95
OCTOBER 2020		
Digital Audio Millivoltmeter	04108191	£9.95
Precision Signal Amplifier	04107191	£7.95
SEPTEMBER 2020		
PE Theremin PSU	AO-0920-01	£5.95
PE Theremin PSU transformer	AO-0920-02	£7.95
Micromite Explore-28	07108191	£6.95
Ultrabrite LED Driver	16109191	£6.95
AUGUST 2020		
Micromite LCD BackPack V3	07106191	£9.95
Steering Wheel Audio Button to Infrared Adaptor	05105191	£7.95
JULY 2020		
AM/FM/CW Scanning HF/VHF RF Signal Generator	04106191	£13.95
Speech Synthesiser with the Raspberry Pi Zero	01106191	£5.95
PE Mini-organ PCB	AO-0720-01	£14.95
PE Mini-organ selected parts	AO-0720-02	£8.95
High-current Solid-state 12V Battery Isolator – control	05106191	£6.95
High-current Solid-state 12V Battery Isolator FET (2oz)	05106192	£9.95
JUNE 2020		
Arduino breakout board – 3.5-inch LCD Display	24111181	£6.95
Six-input Audio Selector main board	01110191	£10.95
Six-input Audio Selector switch panel board	01110192	
MAY 2020		
Ultra-low-distortion Preamplifier Input Selector	01111112	£11.25
Ultra-low-distortion Preamplifier pushbutton Input Selector	01111113	
Universal Regulator	18103111	£7.95
433MHz Wireless Data Repeater	15004191	£8.50
Bridge-mode Adaptor for Amplifier	01105191	£7.95
iCEstick VGA Terminal	02103191	£5.95
Analogue noise with tilt control	AO-0520-01	£7.95
Audio Spectrum Analyser	PM-0520-01	£8.95
APRIL 2020		
Flip-dot Display black coil board	19111181	£14.95
Flip-dot Display black pixels	19111182	
Flip-dot Display black frame	19111183	
Flip-dot Display green driver board	19111184	

PCBs for most recent *PE*/*EPE* constructional projects are available. From the July 2013 issue onwards, PCBs with eight-digit codes have silk screen overlays and, where applicable, are double-sided, have plated-through holes, and solder mask. They are similar to photos in the project articles. Earlier PCBs are likely to be more basic and may not include silk screen overlay, be single-sided, lack plated-through holes and solder mask.

Always check price and availability in the latest issue or online. A large number of older boards are listed for ordering on our website.

In most cases we do not supply kits or components for our projects. For older projects it is important to check the availability of all components before purchasing PCBs.

Back issues of articles are available – see Back Issues page for details.

Double-sided | plated-through holes | solder mask

PROJECT	CODE	PRICE
MARCH 2020		
Diode Curve Plotter	04112181	£10.95
Steam Train Whistle / Diesel Horn Sound Generator	09106181	£8.50
Universal Passive Crossover (one off)	UPC0320	£12.50
FEBRUARY 2020		
Motion-Sensing 12V Power Switch	05102191	£5.95
USB Keyboard / Mouse Adaptor	24311181	£8.50
DSP Active Crossover (ADC)	01106191	£29.95
DSP Active Crossover (DAC) ×2	01106192	
DSP Active Crossover (CPU)	01106193	
DSP Active Crossover (Power/routing)	01106194	
DSP Active Crossover (Front panel)	01106195	
DSP Active Crossover (LCD)	01106196	
JANUARY 2020		
Isolated Serial Link	24107181	£8.50
DECEMBER 2019		
Extremely Sensitive Magnetometer	04101011	£16.75
Four-channel High-current DC Fan and Pump Controller	05108181	£8.75
Useless Box	08111181	£11.50
NOVEMBER 2019		
Tinnitus & Insomnia Killer (Jaycar case – see text)	01110181	£9.95
Tinnitus & Insomnia Killer (Altronics case – see text)	01110182	£9.95
OCTOBER 2019		
Programmable GPS-synced Frequency Reference	04107181	£11.50
Digital Command Control Programmer for Decoders	09107181	£9.95
Opto-isolated Mains Relay (main board)	10107181	£11.50
Opto-isolated Mains Relay (2 × terminal extension board)	10107182	
AUGUST 2019		
Brainwave Monitor	25108181	£12.90
Super Digital Sound Effects Module	01107181	£6.95
Watchdog Alarm	03107181	£8.00
PE Theremin (three boards: pitch, volume, VCA)	PETX0819	£19.50
PE Theremin component pack (see p.56, August 2019)	PETY0819	£15.00
JULY 2019		
Full-wave 10A Universal Motor Speed Controller	10102181	£12.90
Recurring Event Reminder	19107181	£8.00
Temperature Switch Mk2	05105181	£10.45
JUNE 2019		
Arduino-based LC Meter	04106181	£8.00
USB Flexitimer	19106181	£10.45
MAY 2019		
2× 12V Battery Balancer	14106181	£5.95
Deluxe Frequency Switch	05104181	£10.45
USB Port Protector	07105181	£5.95
APRIL 2019		
Heater Controller	10104181	£14.00
MARCH 2019		
10-LED Bargraph Main Board	04101181	£11.25
+Processing Board	04101182	£8.60
FEBRUARY 2019		
1.5kW Induction Motor Speed Controller	10105122	£35.00
NOVEMBER 2018		
Super-7 AM Radio Receiver	06111171	£27.50
OCTOBER 2018		
6GHz+ Touchscreen Frequency Counter	04110171	£12.88
Two 230VAC MainsTimers	10108161	£12.88
	10108162	
SEPTEMBER 2018		
3-Way Active Crossover	01108171	£17.95
Ultra-low-voltage Mini LED Flasher	16110161	£5.95
AUGUST 2018		
Universal Temperature Alarm	03105161	£7.95
Power Supply For Battery-Operated Valve Radios	18108171	£24.95
	18108172	
	18108173	
	18108174	
JULY 2018		
Touchscreen Appliance Energy Meter – Part 1	04116061	£14.95
Automotive Sensor Modifier	05111161	£12.95
JUNE 2018		
High Performance 10-Octave Stereo Graphic Equaliser	01105171	£14.95

For the many pre-2018 PCBs that we stock please see the PE website: www.electronpublishing.com

PE/EPE PCB SERVICE

Order Code	Project	Quantity	Price

Name ....

Address ....

....

Tel ....

Email ....

I enclose payment of £ .... (cheque/PO in **£ sterling only**)

payable to: **Practical Electronics**

Card No ....

Valid From ....Expiry Date ....

Card Security No ....

You can also order PCBs by phone, email or via the shop on our website: www.electronpublishing.com

No need to cut your issue – a copy of this form is just as good!

All prices include VAT and UK p&p. Add £4 per *project* for post to Europe; £5 per *project* outside Europe.

Orders and payment should be sent to:

Practical Electronics, Electron Publishing Ltd
113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU
Tel 01202 880299 Email: shop@electronpublishing.com
On-line Shop: www.epemag.com

Cheques should be made payable to 'Practical Electronics' **(Payment in £ sterling only)**.

NOTE: Most boards are in stock and sent within seven days of receipt of order, please allow up to 28 days delivery if we need to restock.

ESR Electronic Components Ltd

All of our stock is RoHS compliant and CE approved. Visit our well stocked shop for all of your requirements or order on-line.

We can help and advise with your enquiry, from design to construction.

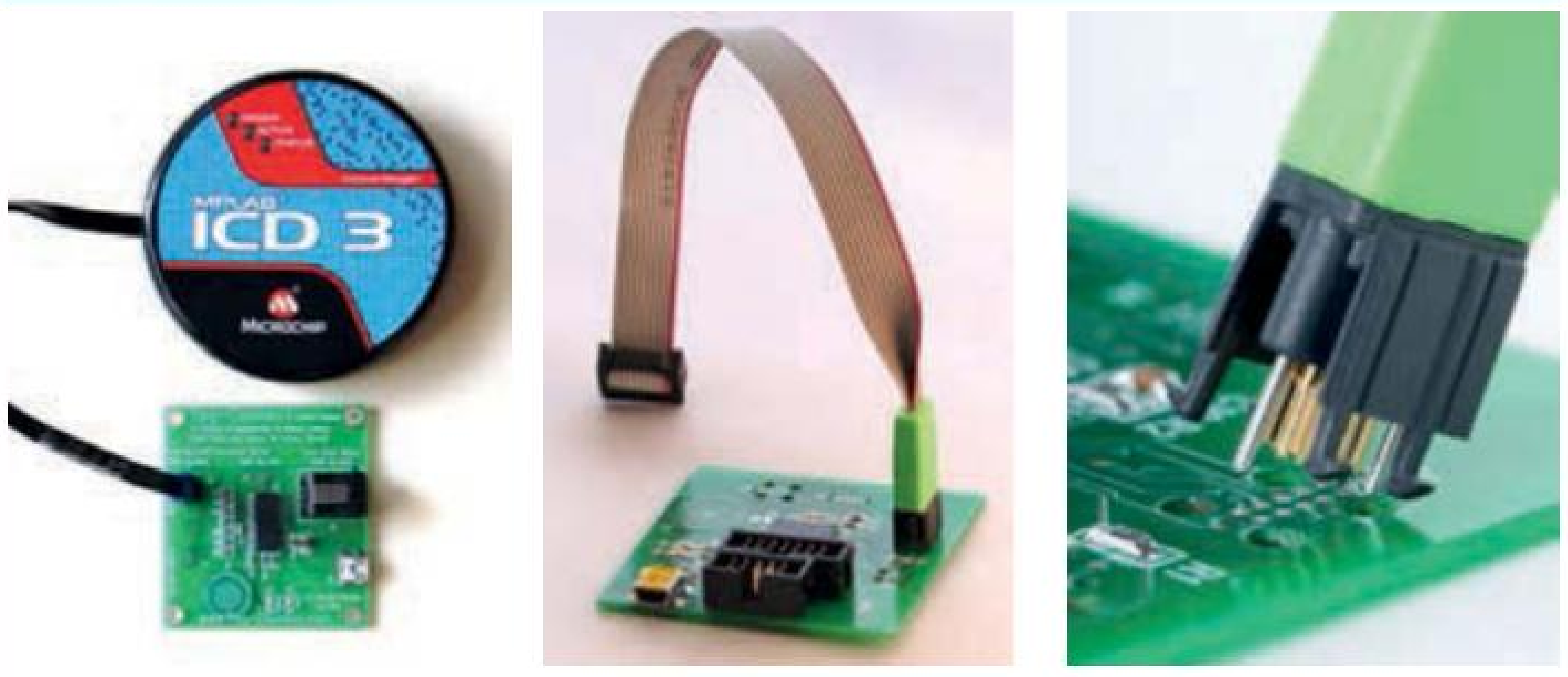
3D Printing • Cable • CCTV • Connectors • Components • Enclosures • Fans • Fuses • Hardware • Lamps • LED's • Leads • Loudspeakers • Panel Meters • PCB Production • Power Supplies • Relays • Resistors • Semiconductors • Soldering Irons • Switches • Test Equipment • Transformers and so much more…

Monday to Friday 08:30 - 17.00, Saturday 08:30 - 15:30

Station Road
Cullercoats
North Shields
Tyne & Wear
NE30 4PQ

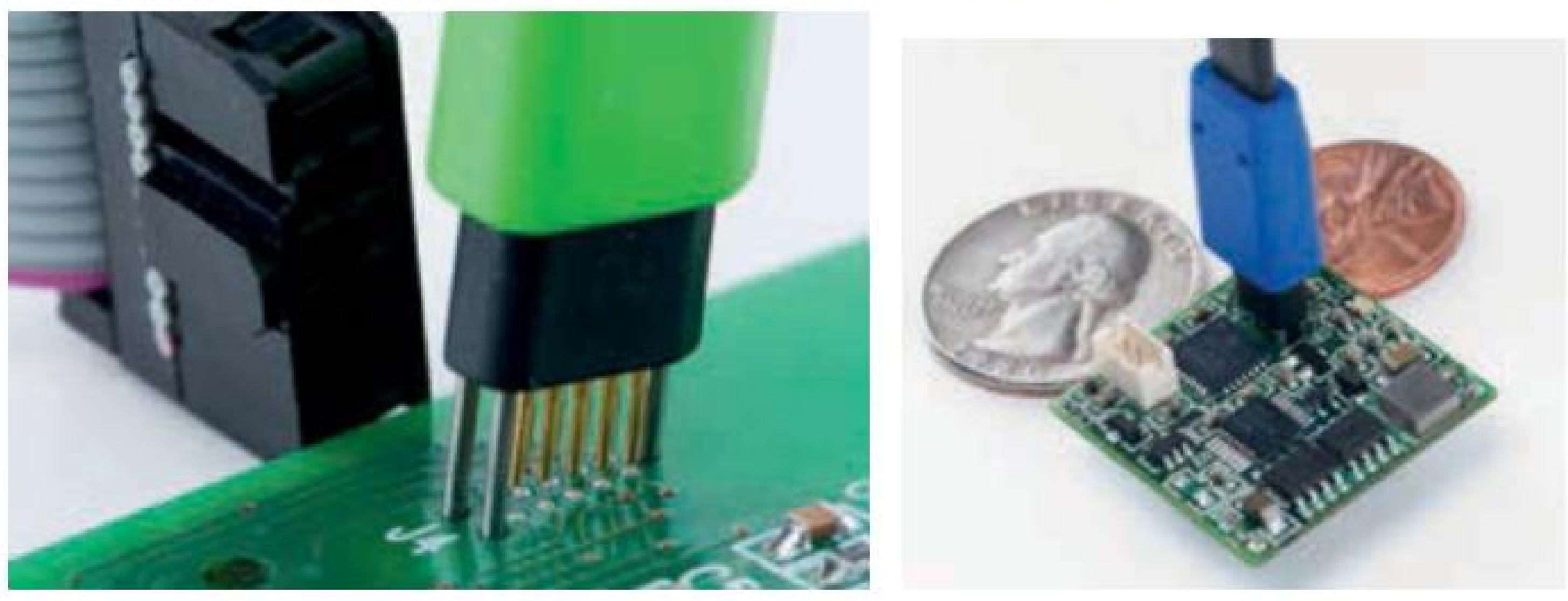
Tel: 0191 2514363 sales@esr.co.uk www.esr.co.uk

Tag-Connect



JTAG Connector Plugs Directly into PCB!!

No Header! No Brainer!



Our patented range of Plug-of-Nails™ spring-pin cables plug directly into a tiny footprint of pads and locating holes in your PCB, eliminating the need for a mating header. Save Cost & Space on Every PCB!!

Solutions for: PIC . dsPIC . ARM . MSP430 . Atmel . Generic JTAG . Altera Xilinx . BDM . C2000 . SPY-BI-WIRE . SPI / IIC . Altium Mini-HDMI . & More

www.PlugOfNails.com

Tag-Connector footprints as small as 0.02 sq. inch (0.13 sq cm)

BACK ISSUES

Practical Electronics

BACK ISSUES – ONLY £5.95

We can supply back issues of PE/EPE by post.

We stock magazines back to 2006, except for the following:

2006 Jan, Feb, Mar, Apr, May, Jul
2007 Jun, Jul, Aug
2008 Aug, Nov, Dec
2009 Jan, Mar, Apr
2010 May, Jun, Jul, Aug, Oct, Nov
2011 Jan
2014 Jan
2018 Jan, Nov, Dec
2019 Jan, Feb, Apr, May, Jun

Issues from Jan 1999 are available on CD-ROM / DVD-ROM

If we do not have a a paper version of a particular issue, then a PDF can be supplied – your email address must be included on your order.

Please make sure all components are still available before commencing any project from a back-dated issue.

CRICKLEWOOD ELECTRONICS

Established 1981

Your best bet since MAPLIN

Chock-a-Block with Stock

Visit: www.cricklewoodelectronics.com

Or phone our friendly knowledgeable staff on **020 8452 0161**

Components • Audio • Video • Connectors • Cables Arduino • Test Equipment etc, etc

Visit our Shop, Call or Buy online at:

www.cricklewoodelectronics.com

020 8452 0161

Visit our shop at:
40-42 Cricklewood Broadway
London NW2 3ET

CLASSIFIED ADVERTISING

Practical Electronics

If you want your advertisements to be seen by the largest readership at the most economical price then our classified page offers excellent value. The rate for semi-display space is £10 (+VAT) per centimetre high, with a minimum height of 2·5cm. All semi-display adverts have a width of 5.5cm. The prepaid rate for classified adverts is 40p (+VAT) per word (minimum 12 words).

Cheques are made payable to 'Practical Electronics'. **VAT must be added**. Advertisements with remittance should be sent to: Practical Electronics, 113 Lynwood Drive, Wimborne, Dorset, BH21 1UU.

Tel 07973518682 Email: pe@electronpublishing.com

For rates and further information on display and classified advertising please contact our Advertisement Manager, Matt Pulzer – see below.

Practical Electronics reaches more UK readers than any other UK monthly hobby electronics magazine. Our sales figures prove it.

We have been the leading monthly magazine in this market for the last twenty-seven years.

BOWOOD ELECTRONICS LTD
Suppliers of Electronic Components
www.bowood-electronics.co.uk
Unit 10, Boythorpe Business Park, Dock Walk, Chesterfield, Derbyshire S40 2QR. Sales: 01246 200 222
Send large letter stamp for Catalogue

MISCELLANEOUS

VALVES AND ALLIED COMPONENTS IN STOCK. Phone for free list. Valves, books and magazines wanted. Geoff Davies (Radio), tel. 01788 574774.

PIC DEVELOPMENT KITS, DTMF kits and modules, CTCSS Encoder and Decoder/Display kits. Visit www.cstech.co.uk

Electrical Industries Charity (EIC)
We help people working in the electrical, electronics and energy community as well as their family members and retirees.

We use workplace programmes that give the industry access to financial grants and a comprehensive range of free and confidential services.
www.electricalcharity.org

electrical industries charity

COAST ELECTRONICS
BREAKOUTS-COMPONENTS-CONTRACT DESIGN-3D PRINTER PARTS-MUSICAL-MICROCONTROLLERS
WWW.COASTELECTRONICS.CO.UK

Andrew Kenny – Qualified Patent Agent
EPO UKIPO USPTO
Circuits Electric Machinery Mechatronics
Web: www.akennypatentm.com
Email: Enquiries@akennypatentm.com
Tel: 0789 606 9725

ADVERTISING INDEX

CRICKLEWOOD ELECTRONICS	70
ESR ELECTRONIC COMPONENTS	70
FLOWCODE	21
HAMMOND ELECTRONICS Ltd	12
JPG ELECTRONICS	72
MICROCHIP	Cover (ii)
PEAK ELECTRONIC DESIGN	Cover (iv)
PICO TECHNOLOGY	37
POLABS D.O.O.	55
QUASAR ELECTRONICS	3
SILICON CHIP	5
STEWART OF READING	63
TAG-CONNECT	70

Advertisement offices

Matt Pulzer

Electron Publishing Ltd
1 Buckingham Road
Brighton
East Sussex BN1 3RA

Tel 07973 518682

Email pe@electronpublishing.com

Web www.electronpublishing.com

For editorial contact details see page 7.

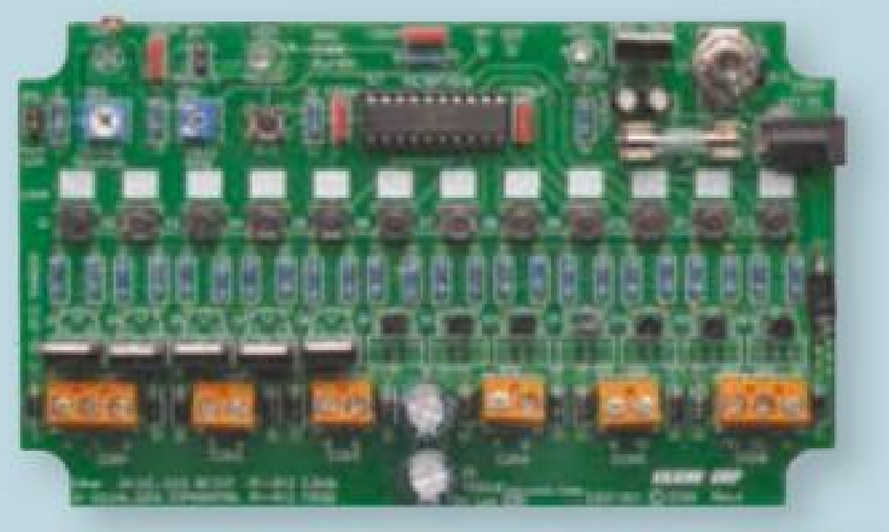
Next Month – in the March issue

Mini Isolated Serial Link

This tiny module (about the size of a postage stamp) provides bidirectional, isolated, full-duplex serial communication. That makes it ideal for when two (or more) boards running from separate supplies need to pass information to each other. It can also carry isolated logic signals.

The Geekcreit LCR-T4 Mini Digital Multi-Tester

It's hard to believe, but you can get a compact digital tester which will identify, check and analyse bipolar transistors, JFETs, MOSFETs, diodes, LEDs and thyristors, resistors, capacitors and inductors for less than the price of a lunch! The Geekcreit LCR-T4 does all of the above and will cost you under a tenner, delivered.



Electronic Wind Chimes – Part 2

In Part 1, we described how our new *Electronic Wind Chimes* worked, and how to build the electronics. Next month, we get to the tricky bit – modifying the wind chime itself so it can be driven by a series of solenoids. Fear not, because we have detailed instructions on how to accomplish this and finish the build by putting it all together and setting up the electronics.

Battery Monitor Logger – Part 2

In Part 1 of our new *Battery Multi-Logger*, we described how it combines the functions of a *Micromite LCD BackPack* along with voltage and current sensing hardware, and power-saving techniques, all on a single PCB. Now we'll go over the construction, testing, setup and calibration procedures so you can build and use it.

PLUS!

All your favourite regular columns from *Audio Out, Cool Beans and Circuit Surgery*, to *Make it with Micromite* and *Net Work*.

On sale 3 February 2022

Content may be subject to change

WELCOME TO JPG ELECTRONICS

Selling Electronics in Chesterfield for 29 Years

Open Monday to Friday 9am to 5:30pm
And Saturday 9:30am to 5pm

- Aerials, Satellite Dishes & LCD Brackets
- Audio Adaptors, Connectors & Leads
- BT, Broadband, Network & USB Leads
- Computer Memory, Hard Drives & Parts
- DJ Equipment, Lighting & Supplies
- Extensive Electronic Components
  - *ICs, Project Boxes, Relays & Resistors*
- Raspberry Pi & Arduino Products
- Replacement Laptop Power Supplies
- Batteries, Fuses, Glue, Tools & Lots more...

T: 01246 211 202
E: sales@jpgelectronics.com
JPG Electronics, Shaw's Row,
Old Road, Chesterfield, S40 2RB
W: www.jpgelectronics.com

Britannia Inn
JPG Electronics
Shaw's Row
Maison Mes Amis
Old Road
Rose & Crown
Johnsons
Old Hall Road
Chatsworth Road
Morrisons
Sparks

JPG Electronics

Retail & Trade Welcome • Free Parking • Google St View Tour: S40 2RB

NEW subscriptions hotline!

Practical Electronics

We have changed the way we sell and renew subscriptions. We now use 'Select Publisher Services' for all print subscriptions – to start a new subscription or renew an existing one you have three choices:

1. Call our NEW print subscription hotline: 01202 087631, or email: pesubs@selectps.com
2. Visit our shop at: www.electronpublishing.com
3. Send a cheque (payable to: 'Practical Electronics') with your details to:

   Practical Electronics Subscriptions, PO Box 6337,
   Bournemouth BH1 9EH, United Kingdom

Remember, we print the date of the last issue of your current subscription in a box on the address sheet that comes with your copy.

Digital subscribers, please call 01202 880299 or visit: www.electronpublishing.com

Published on approximately the first Thursday of each month by Electron Publishing Limited, 1 Buckingham Road, Brighton, East Sussex BN1 3RA. Printed in England by Acorn Web Offset Ltd., Normanton WF6 1TW. Distributed by Seymour, 86 Newman St., London W1T 3EX. Subscriptions UK: £26.99 (6 months); £49.85 (12 months); £94.99 (2 years). EUROPE: airmail service, £30.99 (6 months); £57.99 (12 months); £109.99 (2 years). REST OF THE WORLD: airmail service, £37.99 (6 months); £70.99 (12 months); £135.99 (2 years). Payments payable to 'Practical Electronics', Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom. Email: pesubs@selectps.com. PRACTICAL ELECTRONICS is sold subject to the following conditions, namely that it shall not, without the written consent of the Publishers first having been given, be lent, resold, hired out or otherwise disposed of by way of Trade at more than the recommended selling price shown on the cover, and that it shall not be lent, resold, hired out or otherwise disposed of in a mutilated condition or in any unauthorised cover by way of Trade or affixed to or as part of any publication or advertising, literary or pictorial matter whatsoever.

Did you know our online shop now sells the current issue of *PE* for £5.49 inc. p&p?

PE
Practical Electronics
The UK's premier electronics and computing maker magazine
Circuit Surgery
Exploring op amp input offsets
Audio Out
Vocoder: Driver Amplifier build
KickStart
Using the I²C bus
Make it with Micromite
Using and interfacing the versatile iButton
ARDUINO
Build this handy Arduino-based power supply
Soothing Electronic Wind Chimes
Mastering AC meters
WIN! Microchip WLR089 Xplained Pro Evaluation Kit WIN!
Learn Flowcode Programming
Battery Monitor Logger
PLUS!
Techno Talk – Should we be worried?
Net Work – Electricity generation and streaming radio
Feb 2022 £5.49
www.electronpublishing.com
@practicalelec
practicalelectronics
Practical Electronics
MiniHeart Heartbeat Simulator
192kHz, 24-bit
Jan 2022 £5.49

You read that right! We now sell the current issue of your favourite electronics magazine for exactly the same price as in the High Street, but we deliver it straight to your door – and for UK addresses we pay the postage. No need to journey into town to queue outside the newsagent. Just go to our website, set up an account in 30 seconds, order your magazine and we'll do the rest.

www.electronpublishing.com

PEAK®
electronic design ltd

15% discount!

Use code *PE15* during checkout

limited time only!

ZEN50 Zener Diode Analyser (inc. LEDs, TVSs etc)

Popular product!

The ***Atlas ZEN*** (model ZEN50) is ideal for testing Zeners, VDRs, transient suppressors, LED strings and many other components.

- **Measure Zener Voltage (from 0.00 up to 50.00V!)**
- **Measure Slope Resistance.**
- **Selectable test current: 2mA, 5mA, 10mA and 15mA.**
- **Very low duty cycle to minimise temperature rise.**
- **Continuous measurements.**
- **Single AAA battery (included) with very long battery life.**
- **Gold plated croc clips included.**
- **Can measure forward voltage of LEDs and LED strings too.**

atlas ZEN
model ZEN50
PEAK electronic design ltd
Vz=27.74V @ 15mA
Slope R=53Ω
Zener diode analyser
Zener Voltage
Slope Resistance
Selectable Test Currents
Vz<50V
on - test
scroll - off
warning: not for in-circuit

~~£46.50~~ £39.53
~~£38.75+VAT~~ **with discount!**

LCR45 LCR and Impedance Meter with Auto and Manual modes

Advanced firmware!

The powerful LCR45 meter that not only identifies and measures your passive components (Inductors, Capacitors and Resistors) but also measures complex impedance, magnitude of impedance with phase and admittance too! Auto and Manual test modes allow you to specify the test frequency and component type.

- **Continuous fluid measurements.**
- **Improved measurement resolution: (<0.2µH, <0.2pF).**
- **Test frequencies: DC, 1kHz, 15kHz, 200kHz.**
- **Measure the true impedance of speakers and more.**
- **Great for hobbyists and professionals.**

LCR and impedance meter
PEAK electronic design
Inductor 0.31Ω
@15kHz
L=701.1µH
LCR45
on - menu
enter - off
warning: avoid charge/voltages

~~£91.50~~ £77.78
~~£76.25+VAT~~ **with discount!**

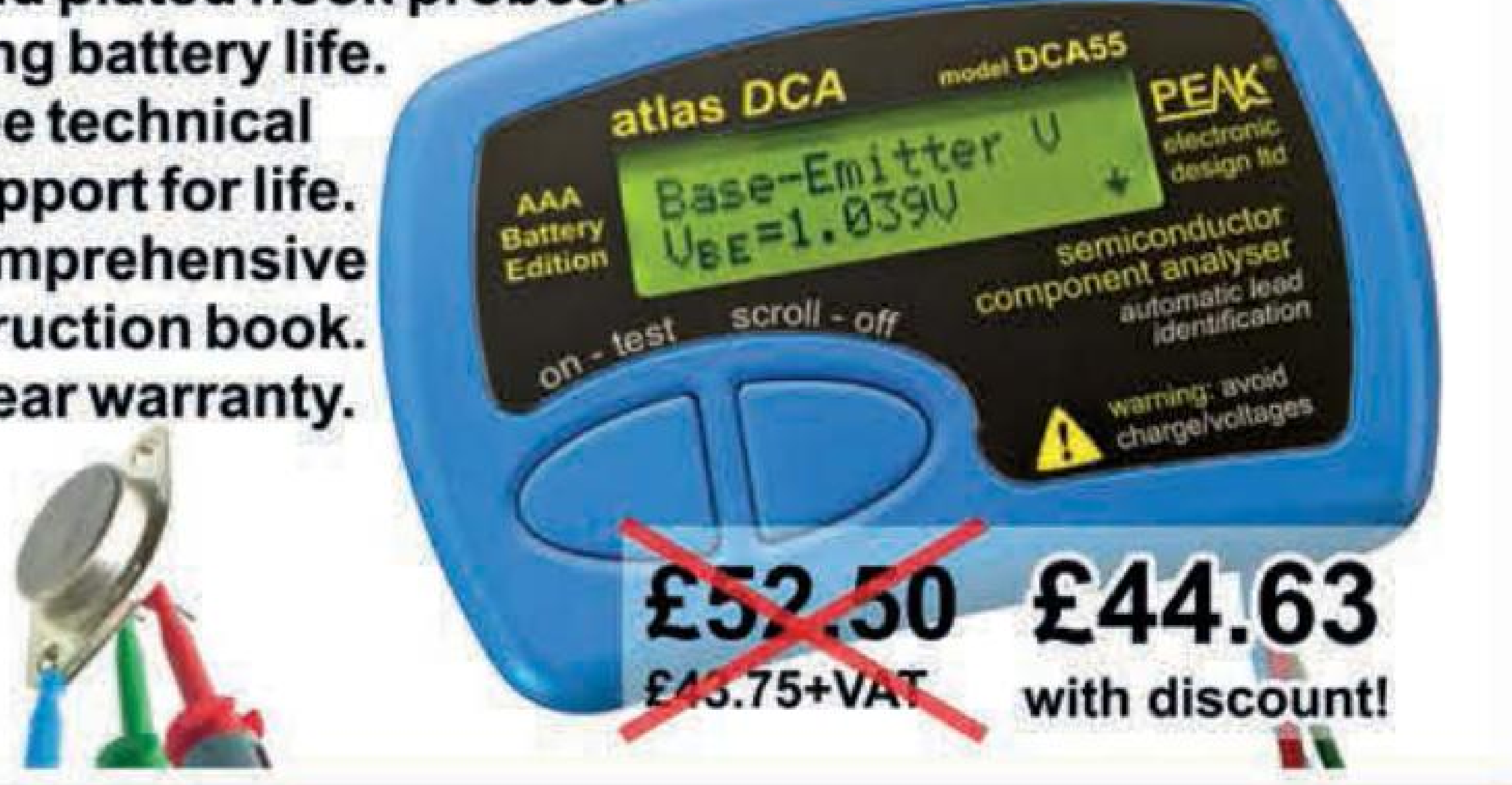
Complex Impedance
Impedance 200kHz
+25.6 +j155.1Ω

DCA55 Semiconductor Analyser - Identify your semi's

Now with backlit display and AAA battery!

Connect any way round to identify the type of component and the pinout! Also measures many parameters including transistor gain, base-emitter voltages, MOSFET thresholds, LED voltages etc. Complete with a comprehensive illustrated user guide. Includes an Alkaline battery so you're ready to go straight away.

- **Transistors (including NPN/PNP, darlingtons, Si & Ge).**
- **Measure hFE, V_{BE} and leakage.**
- **Diodes and LEDs. Measure V_F.**
- **MOSFETs. Measure $V_{GS(th)}$.**
- **Gold plated hook probes.**
- **Long battery life.**
- **Free technical support for life.**
- **Comprehensive instruction book.**
- **2 year warranty.**


atlas DCA
model DCA55
PEAK electronic design ltd
AAA Battery Edition
Base-Emitter V
VBE=1.039V
semiconductor component analyser
automatic lead identification
on - test
scroll - off
warning: avoid charge/voltages

~~£52.50~~ £44.63
~~£43.75+VAT~~ **with discount!**

ESR70 Equivalent Series Resistance and Capacitance Meter

Now with backlit display and AAA battery!

The new improved ESR70 is great for identifying capacitors (especially electrolytics) that are suffering from high ESR. ESR is a great indicator of capacitor condition, especially in applications demanding high ripple currents such as reservoir capacitors, SMPSUs, motor capacitors etc. Includes Alkaline AAA battery and a very comprehensive user guide. The ESR70 will even perform a controlled discharge of your caps!

atlas ESR+
model ESR70
PEAK electronic design ltd
Cap = 22.28µF
ESR = 0.48Ω
ESR, low resistance and capacitance meter
+ Audible Alerts
with automatic discharge function
on - test
off

~~£84.00~~ £71.40
~~£70.00+VAT~~ **with discount!**

It's only possible to show summary specifications here. Please ask if you'd like detailed data. Further information is also available on our website. Product price refunded if you're not happy.

www.peakelec.co.uk
sales@peakelec.co.uk

Atlas House, 2 Kiln Lane, Harpur Hill Business Park
Buxton, Derbyshire, SK17 9JL, UK. Tel: 01298 70012